

ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ

ЕДИНЫЙ ГОСУДАРСТВЕННЫЙ ЭКЗАМЕН

2015



ИНФОРМАТИКА

ОПТИМАЛЬНЫЙ
БАНК ЗАДАНИЙ
ДЛЯ ПОДГОТОВКИ К

ЕГЭ

РУССКИЙ ЯЗЫК
МАТЕМАТИКА
ФИЗИКА
ХИМИЯ
БИОЛОГИЯ
ГЕОГРАФИЯ
ИСТОРИЯ
ОБЩЕСТВОЗНАНИЕ
ЛИТЕРАТУРА
ИНФОРМАТИКА
АНГЛИЙСКИЙ ЯЗЫК
НЕМЕЦКИЙ ЯЗЫК
ФРАНЦУЗСКИЙ ЯЗЫК

ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ

В.Р. Лещинер, С.С. Крылов, А.П. Якушкин

**ОПТИМАЛЬНЫЙ БАНК ЗАДАНИЙ
ДЛЯ ПОДГОТОВКИ К ЕГЭ**

**ЕДИНЫЙ
ГОСУДАРСТВЕННЫЙ
ЭКЗАМЕН
2015**

ИНФОРМАТИКА



Москва
«Интеллект-Центр»
2015

УДК 373.167.1:004

ББК 32.81я721

Л 54

Для создания пособия Федеральным институтом педагогических измерений авторам предоставлено право использования ресурсов открытого банка заданий

Лещинер, В.Р.

Л 54 Оптимальный банк заданий для подготовки к ЕГЭ. Единый государственный экзамен 2015. Информатика. Учебное пособие. / В.Р. Лещинер, С.С. Крылов, А.П. Якушкин. – Москва: Интеллект-Центр, 2015. – 176 с.

ISBN 978-5-00026-118-7

В настоящем пособии представлена система подготовки к сдаче ЕГЭ по информатике, в основе которой лежит знакомство с содержанием экзаменационных заданий, алгоритмами и образцами рассуждения для их правильного выполнения. В нем содержатся общие методические рекомендации по подготовке к экзамену, а также набор типовых тренировочных заданий с методическими указаниями и ответами, что позволяет закрепить полученные знания и подготовиться к сдаче ЕГЭ.

Пособие адресовано старшеклассникам, преподавателям и родителям. Оно поможет школьникам проверить свои знания и умения по предмету, а учителям – оценить степень достижения требований образовательных стандартов отдельными учащимися и обеспечить их целенаправленную подготовку к экзамену.

Издание прошло экспертизу Федерального института педагогических измерений.

УДК 373.167.1:004

ББК 32.81я721

Ответственный за выпуск
Исполнительный директор *О.С. Ильясов*

Редактор *Д.П. Локтионов*
Художественный редактор *Е.Ю. Воробьева*
Компьютерная верстка и макет *Ю.Д. Савченко*

Подписано в печать 04.09.2014 г. Формат 60x84/8.
Печать офсетная. Усл. печ. л. 22,0. Тираж 5000 экз.
Заказ № 1406850.

Издательство «Интеллект-Центр»
125445, Москва, ул. Смольная, д. 24, оф. 712

arvato
BERTELSMANN

Отпечатано в полном соответствии с качеством предоставленного электронного оригинал-макета в ОАО «Ярославский полиграфический комбинат» 150049, Ярославль, ул.Свободы, 97

ISBN 978-5-00026-118-7

© «Интеллект-Центр», 2015

© В.Р. Лещинер, С.С. Крылов,
А.П. Якушкин, 2014

ВВЕДЕНИЕ

Единый государственный экзамен – это одновременно выпускной экзамен за курс среднего (полного) общего образования и вступительный экзамен в учреждения высшего и среднего профессионального образования. Это делает ЕГЭ по информатике экзаменом, ориентированным в первую очередь на выпускников профильных и специализированных классов, собирающихся продолжить свое образование в областях, связанных с информационными технологиями.

Единый государственный экзамен проводится с использованием специально подобранных серий заданий стандартизированной формы – контрольных измерительных материалов (КИМ). Проведение экзамена в такой форме позволяет с достаточной степенью точности установить уровень освоения выпускниками федерального компонента государственного образовательного стандарта среднего (полного) общего образования и, в то же время, уверенно ранжировать их по этому уровню. В связи с этим, идя на государственный экзамен, выпускник должен четко представлять себе уровень освоения предусмотренных учебной программой знаний, умений и навыков, который он стремится показать в зависимости от поставленной цели, а также те типы и модели учебных задач, которые необходимо решать, для того, чтобы показать достижение требуемого уровня. Задача правильно решить все содержащиеся в варианте задания может быть достигнута очень небольшим количеством выпускников, обладающих очень хорошей подготовкой на уровне профильной школы, но для достижения требуемого для поступления в избранный вуз балла экзаменуемым часто не нужно справиться со всеми заданиями экзаменационной работы, достаточно только не сделать ошибок в заданиях, соответствующих требуемому уровню. Поэтому хорошая подготовка к экзамену заключается в стабильном безошибочном решении заданий, проверяющих содержание, освоенное выпускником в процессе обучения в школе.

Данная книга должна помочь выпускникам подготовиться к ЕГЭ по информатике и ИКТ. Составленная научными работниками и преподавателями вузов, имеющими большой опыт разработки диагностических тестов, в том числе заданий контрольно-измерительных материалов ЕГЭ, она призвана осветить полный объем содержания, проверяемого на экзамене, и показать основные типы заданий, которые могут встретиться на экзамене по каждой теме.

Экзаменационная работа 2015 года состоит из двух частей.

Часть 1 содержит 23 задания базового, повышенного и высокого уровней сложности. В этой части собраны задания с краткой формой ответа, подразумевающие самостоятельное формулирование и ввод ответа в виде последовательности символов (чаще всего натурального числа). Задания выполняются на черновике, а ответы заносятся в бланк для ответов № 1. Каждое задание оценивается одним баллом, максимальное количество баллов, которое можно получить за правильное выполнение заданий этой части – 23 балла. Задания проверяются компьютером в автоматическом режиме, путем сравнения ответа с эталонным.

Часть 2 содержит 4 задания, первое из которых повышенного уровня сложности, остальные три задания – высокого уровня сложности. Задания этой части подразумевают запись в произвольной форме развернутого ответа на бланке № 2 (и дополнительных к нему, в случае необходимости). Решения проверяются экспертами региональных предметных комиссий на основе единых для всех участников экзамена критериев. За правильное выполнения задания 24 экзаменуемый может получить до 3 баллов, за задание 25 – до двух баллов, за задание 26 можно заработать до 3 баллов, и, наконец, правильное выполнение задания 27 принесет экзаменуемому 4 балла. В целом правильное выполнение заданий второй части приносит экзаменуемому 12 первичных баллов.

Таким образом, в экзаменационной работе содержится 27 заданий, правильное выполнение которых принесет экзаменуемому 35 первичных баллов. Первичные баллы по определенной формуле переводятся в тестовые баллы по 100-балльной шкале, что позволяет сравнивать результаты по разным предметам с разными шкалами первичных баллов.

В данной книге, в отличие от других пособий для подготовки к ЕГЭ, задачи собраны по темам. По каждой из тем, проверяемых на экзамене, собраны задачи всех известных типов. Учащимся нет необходимости просматривать множество однотипных вариантов в попытке составить представление о разнообразии встречающихся на экзамене задач, которые могут стоять в данной позиции – авторы пособия сделали это за них, собрав задания в тематические блоки.

В работе содержатся задания по 10 разделам курса информатики. На определенные темы заданий больше, на другие темы – меньше. На одну тему в работе могут содержаться несколько заданий различного уровня сложности. Таблица ниже показывает соответствие тем и разделов курса.

№	Название раздела	Число заданий	Номера заданий, уровень сложности	Максимальный первичный балл
1	Информация и её кодирование	4	1 (Б), 9 (Б), 10 (Б), 13 (П)	4
2	Моделирование и компьютерный эксперимент	2	5 (Б), 15 (П)	2
3	Системы счисления	2	4 (Б), 16 (П)	2
4	Основы логики	3	2 (Б), 18 (П), 23 (В)	3
5	Элементы теории алгоритмов	8	6 (Б), 11 (Б), 14 (П), 19 (П), 20(П), 22 (П), 25 (В), 26(В)	11
6	Архитектура компьютеров и компьютерных сетей	1	12 (Б)	1
7	Обработка числовой информации в табличном процессоре	1	7 (Б)	1
8	Технологии поиска и хранения информации	2	3 (Б), 17 (П)	2
9	Программирование	4	8 (Б), 21 (П), 24 (П), 27(В)	9
	Итого:	27		35

Видно, что наибольшим «весом» в итоговом результате обладают три раздела: «Элементы теории алгоритмов», «Программирование» и «Информация и её кодирование». Хорошее знание этих разделов принесет около двух третей итоговых баллов. К первым двум из указанных разделов относятся все задания с развернутым ответом, находящиеся во второй части работы.

В 2015 году контрольно-измерительные материалы (КИМ) ЕГЭ по информатике и ИКТ будут существенно отличаться от экзаменационных вариантов 2012–2014 годов. Прежде всего, в экзаменационной работе не будет заданий с выбором ответа. Это исключит возможность случайного угадывания верного ответа. Следует отметить, что в данной книге собрано довольно много заданий с выбором ответа прежних лет. Их выполнение поможет подготовиться к экзамену 2015 г., так как те же задания могут быть сформулированы в формате, предполагающем краткий ответ. Вместе с тем, для ряда заданий станет невозможным решение методом подстановки ответов. Естественно, что любое из заданий с кратким ответом имеет единственный верный ответ и прямое решение, приводящее к нему.

Следует обратить внимание, что в 2015 г. общее количество заданий в экзаменационной работе сократилось на 5 заданий и максимальный первичный балл за безошибочное выполнение всех заданий также уменьшился с 40 до 35 баллов. Это означает, в частности, что вклад каждого первичного балла в окончательный результат по 100-балльной шкале возрос, возросла и цена возможной ошибки.

Результаты экзамена 2014 г. показали высокую важность хорошей подготовки экзаменуемых к экзамену. Задания, аналогичные тем, что содержатся в открытом банке и были представлены в демонстрационной версии КИМ, были выполнены экзаменуемыми с более высоким результатом, чем новые по форме или содержанию задания. Высокому результату способствует знакомство со структурой работы, представление о возможных типах и форматах заданий на ту или иную тему.

Во всех задачах, содержащих фрагменты программ, они записаны на 4-х языках программирования: Бейсике, Паскале, Си и Алгоритмическом языке. Задание С2 можно также выполнять на естественном языке или построив блок-схему, задача С4 может выполняться на любом известном языке программирования.

За правильное выполнение всей работы можно получить 35 первичных баллов. При пересчете в 100-балльную шкалу они превратятся в 100 тестовых баллов. За правильное выполнение заданий с кратким ответом вы получите 23 первичных балла, а полное и правильное выполнение четырех заданий с развернутым ответом может дать вам 12 первичных баллов.

В 2012–2014 годах сертификат о сдаче ЕГЭ давался тем, кто получил на экзамене по информатике и ИКТ 8 или больше первичных баллов. Это около 90% участников экзамена.

На выполнение экзаменационной работы отводится 235 минут, это чуть менее 4 часов. На выполнение 23 заданий первой части рекомендуется отводить 1,5 часа (90 минут). На выполнение заданий с развернутым ответом рекомендуется отводить около 145 минут. При этом следует знать, что все задания варианта КИМ выдаются экзаменуемому одновременно (в начале экзамена), поэтому участник экзамена вправе самостоятельно распределять время, выполнять задания в любом порядке, возвращаться к уже выполненным заданиям и так далее. Рекомендуется выполнять задания в том порядке, в котором они идут в экзаменационной работе, так как уровень сложности заданий последовательно возрастает. Если экзаменуемый испытывает затруднения при выполнении заданий 21, 22 и 23, их следует пропустить и начать выполнение заданий с развернутым ответом. Только выполнив задания 24, 25 и 26, следует вернуться к заданиям 21–23. Задание 27 наиболее сложное и трудоемкое, к нему следует приступать, выполнив все остальные задания, так как в противном случае может не хватить времени.

Если задание ставит вас в тупик – пропустите его, попытайтесь выполнить следующее по очереди задание. Возвращаться к вызвавшим затруднения заданиям следует только после того, как будут выполнены все задания, не вызывающие затруднений. Учтите, что формулировки заданий могут различаться довольно значительно при одном и том же проверяемом содержании, поэтому важно в первую очередь вчитаться в формулировку, понять, какое содержание она проверяет и каким из разобранных при подготовке к экзамену заданий соответствует, а затем уже начать выполнение задания.

При выполнении заданий первой части следует помнить, что правильным является только ответ, где перечислены все удовлетворяющие условию элементы. Дихотомическая оценка (верно-неверно) заданий с кратким ответом требует полного и исчерпывающего ответа на вопрос. Поэтому ответы, где перечислены не все, а только некоторые отвечающие формулировке значения (числа, комбинации символов и т.п.), будут оценены как неверные, даже если два из трех чисел перечислены верно.

При подготовке к выполнению заданий третьей части работы следует обязательно ознакомиться с указаниями для экспертов по проверке и оцениванию работ. Надо понимать, что проверять ваше решение будет человек, а не компьютер. Это оценивание эксперт проводит, руководствуясь четкими критериями оценки правильности решения, поэтому стоит постараться, чтобы изложение решения было четким, понятным и лаконичным.

И, наконец, важно не только выполнить задание, найти решение и получить ответ, но и правильно, четко, в соответствии с рекомендованным начертанием записать ответ в бланке ответов. Задача машинного распознавания рукописного текста сама по себе достаточно сложна, поэтому в ваших собственных интересах облегчить труд компьютеру, четко следуя образцам начертания цифр и букв, и, тем самым, избежать возможных ошибок распознавания. Линии должны быть непрерывными, буквы и цифры – четкими и контрастными. Именно поэтому бланки ответов следует заполнять черными гелевыми ручками. Также следует помнить, что ответы обрабатываются автоматически, поэтому не существует технической процедуры определить правильность ответа, если он был записан в ненадлежащее поле бланка ответов.

Ответы на задания второй части предназначены для проверки экспертами – специально подготовленными учителями и преподавателями вузов. Они, конечно, в своей жизни проверили много тысяч письменных работ и научились читать любой почерк, но не следует забывать, что они живые люди и, конечно, им читать четкий и понятный текст гораздо удобнее, чем каракули,

испещренные пометками и пометками. Поэтому задания части С выполняются на черновике и переносятся на бланк для записи развернутых ответов только тогда, когда решение полностью готово и надлежащим образом оформлено.

Единый государственный экзамен – комплексное испытание, его результат зависит не только от ваших знаний, умений и навыков, но также от аккуратности, собранности, целеустремленности. Целенаправленная подготовка и хорошая физическая форма в день экзамена – залог успеха.

Лучший способ подготовки к экзамену – изучение информатики. Те, кто хорошо знают предмет, сдают экзамен легко. Тем не менее, стоит прийти на экзамен, хорошо представляя, как будет выглядеть экзаменационная работа. Поэтому посмотрите, пожалуйста, демонстрационную версию экзаменационной работы, размещенную на сайте ФИПИ (www.fipi.ru). Также подготовиться к экзамену вам помогут различные издания, подготовленные специалистами ФИПИ и содержащие примеры заданий, которые использовались на экзаменах в предшествующие годы. Учтите, что демонстрационная версия КИМ только показывает, какое задание МОЖЕТ стоять на этом месте в работе, это совсем не означает, что в работе будет задание абсолютно аналогичное заданию, приведенному в демоверсии. В реальном КИМ задания на каждой позиции будут соответствовать демонстрационному варианту по тематике, уровню сложности и формату ответа. Это не исключает того, что может проверяться какой-то иной аспект темы и задания будут сильно различаться внешне.

Кроме того, четыре позиции в спецификации экзаменационной работы 2015 года (задания 3, 6, 7 и 9) предполагают возможность проверки содержания по двум смежным темам курса. С целью показать разнообразие содержания возможных заданий на этих позициях в демонстрационный вариант были включены задания по двум различным темам в каждой из этих четырех позиций. Следует понимать, что в реальном КИМ задание на каждой из этих позиций будет ровно одно.

Тем не менее, решение заданий прошлых лет может вам очень помочь при подготовке к экзамену, так как вы увидите какие бывают задания, какие элементы содержания предмета проверяются и каким образом это делается. Эта книга призвана помочь вам на основном этапе подготовки к экзамену – при изучении основного материала и тематическом контроле в ходе этого изучения. На этом этапе надо стараться решать задачи разных типов, проверяющие знание данной темы, анализировать возможные ошибки и закреплять умение, решая задачи аналогичного содержания и формата.

Используйте единый государственный экзамен как возможность проверить свои знания, умения, свою волю, свой характер. Готовьтесь к экзамену хорошенько – и благоприятный исход вам гарантирован!

СОДЕРЖАНИЕ КУРСА ИНФОРМАТИКИ И ИКТ В ТЕСТОВЫХ ЗАДАНИЯХ

1. Математические основы информатики

1.1. Кодирование информации

Следует помнить основные положения алфавитного подхода к измерению количества информации.

1) Пусть A – упорядоченное множество из N элементов, тогда для кодирования каждого элемента двоичным кодом, например, путем нумерации в двоичной системе счисления, требуется $\log_2 N$ двоичных разрядов (бит). Объем информации I , содержащейся в сообщении о том, что выбран какой-либо элемент этого множества, равен, соответственно $\log_2 N$ бит. Если N не является целой степенью 2, то число $\log_2 N$ не является целым, и $I = \lceil \log_2 N + 1 \rceil$, т.е. происходит округление в большую сторону. При решении задач, если N не является целым числом, I можно найти как $\log_2 N'$, где N' ближайшая к N степень двойки, такая что $N' > N$.

2) (Следует из предыдущего.) Если некоторый алфавит содержит M символов, то информационный объем одного символа этого алфавита в сообщении равен $\log_2 M$. Для того чтобы найти информационный объем сообщения, состоящего из символов этого алфавита, следует $\log_2 M$ умножить на количество символов в сообщении.

3) С помощью n двоичных разрядов (бит) можно закодировать двоичным кодом все элементы множества мощностью 2^n (т.е. состоящего из 2^n элементов). Информационный объем одного символа, обозначающего элемент данного множества, будет равен n .

Кодирование текста

В1 (демоверсия 2012, в 2013–2014 гг. заданий подобного содержания в КИМ не было)

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной в 20 символов, первоначально записанного в 2-байтном коде Unicode, в 8-битную кодировку КОИ-8. На сколько бит уменьшилась длина сообщения? В ответе запишите только число.

Решение:

После перекодировки из 2-байтного кода в 8-битный каждый символ сообщения стал занимать на 8 бит меньше, а всего символов 20, следовательно длина сообщения уменьшилась на $20 \times 8 = 160$ бит.

Ответ: 160

Задачи для самостоятельного решения

В1.1

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 440 бит. Какова длина сообщения в символах?

В1.2

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной в 30 символов, первоначально записанного в 8-битной кодировке КОИ-8, в 2-байтный код Unicode. На сколько бит увеличилась длина сообщения?

В1.3

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной в 50 символов, первоначально записанного в 2-байтном коде Unicode, в 8-битную кодировку КОИ-8. На сколько байт уменьшилась длина сообщения?

В1.4

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 2-байтном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 200 бит. Какова длина сообщения в символах?

В1.5

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного, первоначально записанного в 8-битной кодировке КОИ-8, в 2-байтный код Unicode. При этом информационное сообщение увеличилось на 320 бит. Какова длина сообщения в символах?

10 (демоверсия 2015)

Все 4-буквенные слова, составленные из букв К, Л, Р, Т, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. КККК
2. КККЛ
3. КККР
4. КККТ

.....

Запишите слово, которое стоит под номером 67.

Решение (1 способ):

Нужно посчитать, сколько слов всего в этом списке (сколько 4-буквенных слов можно составить из 4 различных букв). Так как на каждой из 4 позиций может стоять любая из 4 букв, то количество слов в списке будет равно $4^4 = 256$.

При этом слова под номерами 1-64 будут начинаться на букву К, под номерами 65-128 будут начинаться на букву Л. Под номером 65 стоит слово ЛККК, а под номером 67 – ЛККР.

Решение (2 способ):

Обозначим К = 0, Л = 1, Р = 2, Т = 3. Тогда получается, что список состоит из чисел в системе счисления с основанием 4, количество значащих цифр в которых не превышает 4.

Таких чисел $4^4 = 256$. При этом число 0000 в списке пронумеровано 1, 0001 – 2, а под номером 67 будет стоять число, равное 66_{10} , которое и нужно будет перевести в четверичную систему счисления.

Разложим 66 на степени четверки ($4^3=64$, $4^2=16$, $4^1=4$, $4^0=1$): $66 = 1 \cdot 64 + 0 \cdot 16 + 0 \cdot 4 + 2 \cdot 1$

Выписав цифры, получим $66_{10} = 1002_4$. Вспомнив начальные обозначения, запишем искомого слово: ЛККР.

Ответ: ЛККР

Первый способ более быстрый в некоторых случаях, но второй способ более надежный и универсальный.

10 (В4 демоверсии 2012)

Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. АААОА

.....

Запишите слово, которое стоит на 240-м месте от начала списка.

Решение (1 способ):

Нужно посчитать, сколько слов всего в этом списке (сколько 5-буквенных слов можно составить из 3 различных букв). Так как на каждой из 5 позиций может стоять любая из 3 букв, то количество слов в списке будет равно $3^5 = 243$.

Значит последнее слово УУУУУ стоит на 243-м месте. На 242-м месте стоит УУУУО, на 241-м – УУУУА, а на 240-м – УУУОУ.

Решение (2 способ):

Обозначим А = 0, О = 1, У = 2, тогда получается, что список состоит из чисел в троичной системе счисления, количество значащих цифр в которых не превышает 5.

Таких чисел $3^5 = 243$. При этом число 00000 в списке пронумеровано 1, 00001 – 2, а под номером 240 будет стоять число, равное 239_{10} , которое и нужно будет перевести в троичную систему счисления.

Разложим 240 на степени тройки ($3^4=81$, $3^3=27$, $3^2=9$, $3^1=3$, $3^0=1$): $239 = 2 \cdot 81 + 2 \cdot 27 + 2 \cdot 9 + 1 \cdot 3 + 2 \cdot 1$

Выписав троичные цифры, получим $239_{10} = 22212_3$. Вспомнив начальные обозначения, запишем искомое слово: УУУОУ.

Ответ: УУУОУ

10 (В4 демоверсии 2013)

Азбука Морзе позволяет кодировать символы для сообщений по радиосвязи, задавая комбинацию точек и тире. Сколько различных символов (цифр, букв, знаков пунктуации и т. д.) можно закодировать, используя код азбуки Морзе длиной не менее четырех и не более пяти сигналов (точек и тире)?

Решение:

В основе решения лежит известная формула $P=a^k$, где P-количество слов, которые можно записать с помощью k символов алфавита мощностью a. Так как символы в азбуке Морзе имеют переменную длину, то для ответа на поставленный вопрос надо сложить два числа: $2^4+2^5=16+32=48$.

Ответ: 48

Задачи для самостоятельного решения**10.1**

Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. АААОА

.....

Запишите слово, которое стоит на **238-м месте** от начала списка.

10.2

Все 5-буквенные слова, составленные из букв А, О, У, Я, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. ААААЯ
5. АААОА

.....

Запишите слово, которое стоит на **1020-м месте** от начала списка.

10.3

Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. АААОА

.....

Запишите слово, которое стоит под номером **61**.

10.4

Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. АААОА

.....

Запишите слово, которое стоит на **83-м месте** от начала списка.

10.5

Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. АААОА

.....

Запишите слово, которое стоит на **195-м месте** от начала списка.

10.6

Для передачи аварийных сигналов договорились использовать специальные цветные сигнальные ракеты, запускаемые последовательно. Одна последовательность ракет – один сигнал; в каком порядке идут цвета – существенно. Какое количество различных сигналов можно передать при помощи запуска ровно четырёх таких сигнальных ракет, если в запасе имеются ракеты четырёх различных цветов (ракет каждого вида неограниченное количество, цвет ракет в последовательности может повторяться)?

10.7

Световое табло состоит из трёх светящихся элементов, каждый из которых может светиться одним из пяти различных цветов. Каждая комбинация из трёх цветов кодирует определённый сигнал. Сколько различных сигналов можно передать при помощи табло при условии, что все элементы должны светиться?

10.8

Для передачи аварийных сигналов договорились использовать специальные цветные сигнальные ракеты, запускаемые последовательно. Одна последовательность ракет – один сигнал; в каком порядке идут цвета – существенно. Какое количество различных сигналов можно передать при помощи запуска ровно пяти таких сигнальных ракет, если в запасе имеются ракеты четырёх различных цветов (ракет каждого вида неограниченное количество, цвет ракет в последовательности может повторяться)?

10.9

Все 4-буквенные слова, составленные из букв К, Л, Р, Т, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. КККК
2. КККЛ
3. КККР
4. КККТ

.....

Запишите слово, которое стоит под номером 131.

13 (демоверсия 2015)

При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из 15 символов и содержащий только символы Ш, К, О, Л, А (таким образом, используется 5 различных символов). Каждый такой пароль в компьютерной системе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит).

Укажите объём памяти в байтах, отводимый этой системой для записи 30 паролей. В ответе запишите только число, слово "байт" писать не нужно.

Решение:

Количество различных символов $N = 5$.

$$N' = 8 = 2^3.$$

$I = \log_2 N' = 3$ – это минимальное количество бит для кодирования одной буквы.

Поскольку всего символов в пароле 15, то объём места, требуемого для хранения одного такого пароля равен $15 \times 3 = 45$ бит.

$5 \text{ байт} < 45 \text{ бит} < 6 \text{ байт}$, следовательно минимально возможное количество байт для хранения одного пароля равно 6.

Поскольку всего паролей 30, то необходимый объём памяти составляет $30 \times 6 = 180$ байт.

Ответ: 180.

13 (A11 демоверсии 2012 г.)

Для регистрации на сайте некоторой страны пользователю требуется придумать пароль. Длина пароля – ровно 11 символов. В качестве символов используются десятичные цифры и 12 различных букв местного алфавита, причем все буквы используются в двух начертаниях: как строчные, так и заглавные (регистр буквы имеет значение!).

Под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов.

Определите объём памяти, который занимает хранение 60 паролей.

- 1) 540 байт 2) 600 байт 3) 660 байт 4) 720 байт

Решение:

Количество различных символов $N = 10 + 12 + 12 = 34$ (10 цифр, 12 строчных букв и 12 заглавных букв).

$$N' = 64 = 2^6.$$

$I = \log_2 N' = 6$ – это минимальное количество бит для кодирования одной буквы.

Поскольку всего символов в пароле 11, то объём места, требуемого для хранения одного такого пароля равен $11 \times 6 = 66$ бит.

8 байт < 66 бит < 9 байт, следовательно минимально возможное количество байт для хранения одного пароля равно 9.

Поскольку всего паролей 60, то необходимый объем памяти составляет $60 \times 9 = 540$ байт.

Ответ: 1.

Задачи для самостоятельного решения

13.1

В некоторой стране автомобильный номер длиной 6 символов составляют из заглавных букв (используются только 20 различных букв) и десятичных цифр в любом порядке. Каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит).

Определите объем памяти, отводимый этой программой для записи 30 номеров.

- 1) 90 байт 2) 120 байт 3) 180 байт 4) 150 байт

13.2

Для регистрации на сайте некоторой страны пользователю требуется придумать пароль. Длина пароля – ровно 8 символов. В качестве символов используются десятичные цифры и 15 различных букв местного алфавита, причем все буквы используются в двух начертаниях: как строчные, так и заглавные (регистр буквы имеет значение!).

Под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов.

Определите объём памяти, который занимает хранение 80 паролей.

- 1) 320 байт 2) 400 байт 3) 560 байт 4) 480 байт

13.3

В некоторой стране автомобильный номер длиной 10 символов составляют из заглавных букв (используются только 18 различных букв) и десятичных цифр в любом порядке.

Каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит).

Определите объем памяти, отводимый этой программой для записи 100 номеров.

- 1) 500 байт 2) 600 байт 3) 700 байт 4) 800 байт

13.4

Для регистрации на сайте некоторой страны пользователю требуется придумать пароль. Длина пароля – ровно 9 символов. В качестве символов используются десятичные цифры и 28 различных букв местного алфавита, причем все буквы используются в двух начертаниях: как строчные, так и заглавные (регистр буквы имеет значение!).

Под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов.

Определите объём памяти, который занимает хранение 40 паролей.

- 1) 200 байт 2) 240 байт 3) 280 байт 4) 320 байт

13.5

Для регистрации на сайте некоторой страны пользователю требуется придумать пароль. Длина пароля – ровно 11 символов. В качестве символов используются десятичные цифры и 10 различных букв местного алфавита, причем все буквы используются в двух начертаниях: как строчные, так и заглавные (регистр буквы имеет значение!).

Под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов.

Определите объём памяти, который занимает хранение 50 паролей.

- 1) 300 байт 2) 350 байт 3) 400 байт 4) 550 байт

13.6

В велокроссе участвуют 56 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Какой объём памяти будет использован устройством, когда все спортсмены прошли промежуточный финиш?

- 1) 56 бит 2) 280 бит 3) 42 байт 4) 56 байт

13.7

При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только цифры и буквы У, Ч, И, Т, Е, Л, Ъ (таким образом, используется 17 различных символов). Каждый такой пароль в компьютерной системе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит).

Определите объём памяти (в байт), отводимый этой системой для записи 40 паролей. *В ответе запишите только число.*

Неравномерное кодирование

Неравномерным называется способ кодирования, при котором разные символы алфавита кодируются битовой последовательностью различной длины. Этот код может оказаться более экономным в случае, если определенные символы алфавита встречаются в тексте часто, а некоторые – достаточно редко. Тогда часто встречающиеся символы следует кодировать короткими кодовыми последовательностями, а редко встречающиеся – длинными. На практике чаще используется равномерное кодирование, но для некоторых видов алгоритмов знание принципов неравномерного кодирования может оказаться полезным.

При решении задач на неравномерное кодирование необходимо знать, когда код является однозначно декодируемым. Для этого введем понятие префиксного кода.

Префиксным называется код, не имеющий ни одного кодового слова, которое было бы префиксом (началом) любого другого кодового слова данного кода. Если код является префиксным, то любая последовательность кодовых слов всегда только одним образом разделяема на отдельные из них.

1 (А9 демоверсии 2012 г.)

Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код:

А–1, Б–000, В–001, Г–011. Укажите, каким кодовым словом должна быть закодирована буква Д. Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

- 1) 00 2) 01 3) 11 4) 010

Решение:

Вариантов кодового слова, которым может быть закодирована буква Д, много, поэтому проверим предложенные ответы, начиная с самых коротких:

- 1) 00 – является префиксом слова 000 (буква Б), отсюда, например, входящую последовательность 000000 можно декодировать двумя различными способами: 00 00 00 (ДДД) и 000 000 (ББ). Этот вариант ответа не подходит;
- 2) 01 – является префиксом слова 011 (буква Г). Входящую последовательность 011 можно декодировать двумя различными способами: 011 (Г) и 01 1 (ДА). Этот вариант ответа также не подходит;
- 3) 11 – не является префиксом ни одного из слов, но при этом кодовое слово 1 (А) является префиксом слова 11 (Д). Входящую последовательность 11 можно декодировать двумя различными способами: 11 (Д) и 1 1 (АА). Этот вариант ответа не подходит;
- 4) 010 – не является префиксом ни одного из слов и, при этом, ни одно из слов кода не является префиксом слова 010.

Ответ: 4.

При кодировании часто используются избыточные коды, рассчитанные на возможность исправления единичных ошибок, которые могут возникать при передаче сообщения. В КИМ ЕГЭ 2013 года задание А9 имело следующий вид:

11 (ОС¹ А9 2013 г.)

Для передачи данных по каналу связи используется 5-битовый код. Сообщение содержит только буквы А, Б и В, которые кодируются следующими кодовыми словами:

А – 10010, Б – 11111, В – 00101.

При передаче возможны помехи. Однако некоторые ошибки можно попытаться исправить. Любые два из этих трёх кодовых слов отличаются друг от друга не менее чем в трёх позициях. Поэтому если при передаче слова произошла ошибка не более чем в одной позиции, то можно сделать обоснованное предположение о том, какая буква передавалась. (Говорят, что «код исправляет одну ошибку».) Например, если получено кодовое слово 01111, считается, что передавалась буква Б. (Отличие от кодового слова для Б только в одной позиции, для остальных кодовых слов отличий больше.) Если принятое кодовое слово отличается от кодовых слов для букв А, Б, В более чем в одной позиции, то считается, что произошла ошибка (она обозначается 'х').

Получено сообщение 10000 10101 11011 10111. Декодируйте это сообщение – выберите правильный вариант.

- | | | | |
|---------|---------|---------|---------|
| 1) АВББ | 2) АВхБ | 3) хххх | 4) АхББ |
|---------|---------|---------|---------|

Решение:

Для решения этого задания следует применить описанный алгоритм по отношению к каждому кодовому слову в сообщении:

- 1) 10000 отличается от 10010 ровно в одной (четвертой) позиции, поэтому его можно интерпретировать как букву А;
- 2) 10101 отличается от 00101 ровно в одной (первой) позиции, поэтому его можно интерпретировать как букву В;
- 3) 11011 отличается от 11111 ровно в одной (третьей) позиции, поэтому его можно интерпретировать как букву Б;
- 4) 10111 отличается от 11111 ровно в одной (второй) позиции, поэтому его можно интерпретировать как букву Б.

Ответ: 1.

¹ Здесь и далее пометка ОС означает, что задачи взяты из открытого сегмента федеральной базы тестовых заданий (www.fipi.ru).

Задачи для самостоятельного решения

1.1

Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А–011, Б–11, В–001, Г–010. Укажите, каким кодовым словом должна быть закодирована буква Д. Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

- 1) 1 2) 10 3) 01 4) 00

1.2

Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А–111, Б–01, В–10, Г–001. Укажите, каким кодовым словом должна быть закодирована буква Д. Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

- 1) 11 2) 00 3) 110 4) 010

1.3

Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А–100, Б–110, В–111, Г–0. Укажите, каким кодовым словом должна быть закодирована буква Д. Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

- 1) 101 2) 11 3) 001 4) 10

1.4

Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А–00, Б–1, В–010, Г–0111. Укажите, каким кодовым словом должна быть закодирована буква Д. Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

- 1) 0 2) 001 3) 101 4) 0110

1.5

Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А–01, Б–00, В–10, Г–110. Укажите, каким кодовым словом должна быть закодирована буква Д. Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

- 1) 001 2) 11 3) 010 4) 111

1.6

Для передачи данных по каналу связи используется 5-битовый код. Сообщение содержит только буквы А, Б и В, которые кодируются следующими кодовыми словами:

А – 11000, Б – 00010, В – 10101.

При передаче возможны помехи. Однако некоторые ошибки можно попытаться исправить. Любые два из этих трёх кодовых слов отличаются друг от друга не менее чем в трёх позициях. Поэтому если при передаче слова произошла ошибка не более чем в одной позиции, то можно сделать обоснованное предположение о том, какая буква передавалась. (Говорят, что «код исправляет одну ошибку».) Например, если получено кодовое слово 00000, считается, что передавалась буква Б. (Отличие от кодового слова для Б только в одной позиции,

для остальных кодовых слов отличий больше.) Если принятое кодовое слово отличается от кодовых слов для букв А, Б, В более чем в одной позиции, то считается, что произошла ошибка (она обозначается 'х').

Получено сообщение 11110 10111 10010 10000. Декодируйте это сообщение – выберите правильный вариант.

- 1) хххх 2) АВБА 3) ххБА 4) хВБА

Прием и передача информации

В задачах по этой теме обычно требуется вычислять либо время передачи определенного объема информации при заданной пропускной способности канала связи, либо объем информации, который теоретически возможно передать через канал определенной пропускной способности за заданное время.

9 (В10 демоверсии 2012 г.)

У Кати есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{20} бит в секунду. У Сергея нет скоростного доступа в Интернет, но есть возможность получать информацию от Кати по телефонному каналу со средней скоростью 2^{13} бит в секунду. Сергей договорился с Катей, что она скачает для него данные объемом 9 Мбайт по высокоскоростному каналу и ретранслирует их Сергею по низкоскоростному каналу.

Компьютер Кати может начать ретрансляцию данных не раньше, чем им будут получены первые 1024 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Катей данных до полного их получения Сергеем? В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Решение:

Поскольку скорость телефонного канала между Сергеем и Катей меньше скорости доступа к сети Интернет Кати, то скорость передачи данных будет постоянна до тех пор, пока не скачается необходимый объем данных. В связи с этим, надо посчитать время загрузки Катей первых 1024 Кбайт по радиоканалу и сложить это время со временем передачи 9 Мбайт по телефонному каналу.

- 1) 1024 Кбайт = $2^{10} \cdot 2^{10}$ байт = 2^{23} бит. Такой объем информации будет качаться $2^{23} : 2^{20} = 8$ секунд;
- 2) Далее 9 Мбайт = $9 \cdot 2^{10} \cdot 2^{10}$ байт = $9 \cdot 2^{23}$ бит. Такой объем информации по низкоскоростному каналу будет качаться $9 \cdot 2^{23} : 2^{13} = 9 \cdot 2^{10} = 9216$ секунд;
- 3) Всего получаем $8 + 9216 = 9224$ секунды.

Ответ: 9224

9 (В10 демоверсии 2013 г.)

Документ объемом 20 Мбайт можно передать с одного компьютера на другой двумя способами.

- А. Сжать архиватором, передать архив по каналу связи, распаковать.
Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и насколько, если:

- средняя скорость передачи данных по каналу связи составляет 2^{20} бит в секунду;
- объём сжатого архиватором документа равен 20% исходного;
- время, требуемое на сжатие документа, – 5 секунд, на распаковку – 1 секунда?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единиц измерения «секунд», «сек.», «с.» к ответу добавлять не нужно.

Решение:

Сначала следует вычислить время передачи документа по указанному каналу способом Б (без сжатия).

$20 \text{ Мбайт} = 20 \cdot 2^{20} \text{ байт} = 20 \cdot 2^{23} \text{ бит}$. Такой объем информации будет качаться $20 \cdot 2^{23} : 2^{20} = 20 \cdot 8 = 160$ секунд;

Аналогично вычисляем время передачи сжатого документа, объем которого в 5 раз меньше:

$4 \text{ Мбайт} = 4 \cdot 2^{20} \text{ байт} = 4 \cdot 2^{23} \text{ бит}$. Такой объем информации будет качаться $4 \cdot 2^{23} : 2^{20} = 4 \cdot 8 = 32$ секунды;

Добавляем к этому времени 6 секунд на архивацию и распаковку, получаем 38 секунд.

Осталось только вычислить разность между полученными значениями: $160 - 38 = 122$.

Способ А на 122 секунды быстрее.

Ответ: А122

Задачи для самостоятельного решения

9.1

У Васи есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{18} бит в секунду. У Алексея нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по телефонному каналу со средней скоростью 2^{15} бит в секунду. Алексей договорился с Васей, что она скачает для него данные объемом 4 Мбайт по высокоскоростному каналу и ретранслирует их Алексею по низкоскоростному каналу.

Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 512 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Алексеем?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

9.2

У Пети есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{21} бит в секунду. У Вани нет скоростного доступа в Интернет, но есть возможность получать информацию от Пети по телефонному каналу со средней скоростью 2^{16} бит в секунду. Ваня договорился с Петей, что она скачает для него данные объемом 5 Мбайт по высокоскоростному каналу и ретранслирует их Ване по низкоскоростному каналу.

Компьютер Пети может начать ретрансляцию данных не раньше, чем им будут получены первые 2024 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Петей данных до полного их получения Ваней?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

9.3

У Лены есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{19} бит в секунду. У Сергея нет скоростного доступа в Интернет, но есть возможность получать информацию от Лены по телефонному каналу со средней скоростью 2^{14} бит в секунду. Сергей договорился с Леной, что она скачает для него данные объемом 7 Мбайт по высокоскоростному каналу и ретранслирует их Сергею по низкоскоростному каналу.

Компьютер Лени может начать ретрансляцию данных не раньше, чем им будут получены первые 128 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Леной данных до полного их получения Сергеем?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

9.4

У Жени есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{18} бит в секунду. У Саши нет скоростного доступа в Интернет, но есть возможность получать информацию от Жени по телефонному каналу со средней скоростью 2^{17} бит в секунду. Саша договорился с Женей, что она скачает для него данные объемом 6 Мбайт по высокоскоростному каналу и ретранслирует их Саше по низкоскоростному каналу.

Компьютер Жени может начать ретрансляцию данных не раньше, чем им будут получены первые 1024 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Женей данных до полного их получения Сашей?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

9.5

У Светы есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{20} бит в секунду. У Гоши нет скоростного доступа в Интернет, но есть возможность получать информацию от Светы по телефонному каналу со средней скоростью 2^{14} бит в секунду. Гоша договорился со Светой, что она скачает для него данные объемом 4 Мбайт по высокоскоростному каналу и ретранслирует их Гоше по низкоскоростному каналу.

Компьютер Светы может начать ретрансляцию данных не раньше, чем им будут получены первые 3 Мбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Светой данных до полного их получения Гошей?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

9.6

Документ объемом 16 Мбайт можно передать с одного компьютера на другой двумя способами.

А. Сжать архиватором, передать архив по каналу связи, распаковать.

Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и насколько, если:

- средняя скорость передачи данных по каналу связи составляет 2^{21} бит в секунду;
- объём сжатого архиватором документа равен 25% исходного;
- время, требуемое на сжатие документа, – 12 секунд, на распаковку – 3 секунды?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единицы измерения «секунд», «сек.», «с.» к ответу добавлять не нужно.

9.7

Документ объемом 12 Мбайт можно передать с одного компьютера на другой двумя способами.

А. Сжать архиватором, передать архив по каналу связи, распаковать.

Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и насколько, если:

- средняя скорость передачи данных по каналу связи составляет 2^{21} бит в секунду;
- объём сжатого архиватором документа равен 75% исходного;
- время, требуемое на сжатие документа, – 13 секунд, на распаковку – 3 секунды?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единицы измерения «секунд», «сек.», «с.» к ответу добавлять не нужно.

1.2. Системы счисления

Двоичная система счисления

Для перевода чисел из двоичной системы в десятичную, и обратно, полезно выучить наизусть таблицу первых десяти степеней двойки:

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

Переведем число 10101010 из двоичной системы счисления в десятичную. Пронумеруем его цифры справа налево, подпишем под ненулевыми разрядами соответствующие степени двойки и просуммируем их:

1	0	1	0	1	0	1	0
7	6	5	4	3	2	1	0
128		32		8		2	

$$10101010_2 = 2+8+32+128 = 170_{10}$$

Для перевода из десятичной в двоичную систему надо разложить данное число на степени двойки методом вычитания старшей степени. Например, $170 = 128+42 = 1 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 10 = 1 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 2 = 1 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$.

Выписав двоичные цифры, получаем равенство $170_{10} = 10101010_2$. Обратите внимание, что очень важно довести выписывание степеней до самого конца, не потеряв ни одной цифры, так как количество нулей в **конце числа** имеет большое значение! (Действительно, каждый понимает, что числа 100_{10} (сто) и 10_{10} (десять) различаются на порядок. То же самое верно и по отношению к двоичным числам: 100_2 (четыре) и 10_2 (два) различаются в два раза.)

Пример:

Сколько единиц в двоичной записи числа 67?

- 1) 1 2) 2 3) 3 4) 4

Решение:

Разложим 67 на степени двойки, начиная со старшей (64):

$$67 = 1 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 1000011_2.$$

Ответ: 3.

Задачи для самостоятельного решения

4.1

Сколько единиц в двоичной записи числа 64?

- 1) 1 2) 2 3) 3 4) 5

4.2

Сколько единиц в двоичной записи числа 513?

- 1) 1 2) 2 3) 3 4) 9

4.3

Сколько значащих нулей в двоичной записи числа 127?

- 1) 1 2) 7 3) 0 4) 8

4.4

Сколько значащих нулей в двоичной записи числа 513?

- 1) 1 2) 0 3) 9 4) 8

4.5

Сколько единиц в двоичной записи числа 31?

- 1) 1 2) 2 3) 3 4) 5

4.6

Сколько значащих нулей в двоичной записи числа 44?

- 1) 1 2) 2 3) 3 4) 9

4.7

Сколько единиц в двоичной записи шестнадцатеричного числа FF1₁₆?**Представление числовой информации. Системы счисления**

При записи числа в позиционной системе счисления, оно обозначается с помощью ряда цифр. «Вклад» каждой цифры в число определяется местом, где она находится, и равен значению цифры, умноженному на основание системы счисления в степени, равной номеру цифры (нумерация ведется справа и начинается с нуля). Пусть

$a_n a_{n-1} a_{n-2} \dots a_1 a_0$ — запись числа A , a_i — цифры, p — основание системы счисления, тогда

$$A = a_n \times p^n + a_{n-1} \times p^{n-1} + a_{n-2} \times p^{n-2} + \dots + a_1 \times p^1 + a_0 \times p^0$$

Так, в числе 1987₁₀ цифра 1 обозначает одну тысячу (третья степень), 9 — девять сотен (вторая степень), 8 — восемь десятков (первая степень) и 7 — семь единиц (нулевая степень). Т.е. $1987_{10} = 1 \times 10^3 + 9 \times 10^2 + 8 \times 10^1 + 7 \times 10^0 = 1000 + 900 + 80 + 7$. Аналогично, число 1010₂ = $= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 10_{10}$

16 (ОС)

В системе счисления с некоторым основанием число 12₁₀ записывается в виде 110. Укажите это основание.

Решение:

Обозначим искомое основание n . Исходя из правил записи чисел в позиционных системах счисления $110_n = n^2 + n^1 + 0$. Составим уравнение: $n^2 + n = 12$ и найдем натуральный корень (3). Второй, отрицательный, корень (−4) квадратного уравнения нам не подходит, так как основание системы счисления, по определению, натуральное число, большее единицы. Полученный ответ проверим подстановкой: $9 + 3 + 0 = 12$.

Существует еще другой вариант решения, основанный на простом подборе. Пусть наше число имеет основание n , тогда оно записывается в виде $110_n = n^2 + n^1 + 0$. Будем подставлять в качестве основания различные натуральные числа, начиная с 2. При $n=2$ получим $110_2 = 5_{10}$, при $n=3$ получим $110_3 = 12_{10}$, то есть искомое решение. Ясно, что при $n>3$ мы будем получать большие числа, например, при $n=4$ получим $110_4 = 20_{10}$.

Ответ: 3.**Пример (ОС):**

Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 17₁₀ оканчивается на 2.

Решение:

Последняя цифра в записи числа представляет собой остаток от деления числа на основание системы счисления. $17 - 2 = 15$. Найдем делители числа 15 — это числа 3, 5, 15. Проверим свой ответ тем, что запишем число 17 в указанных системах счисления.

$$17_{10} = 122_3 = 32_5 = 12_{15}$$

Ответ: 3, 5, 15.**Пример:**

Сколько единиц в двоичной записи значения выражения $4^{2015} - 2^{2014} + 3$?

Решение:

Число 2^k (4^k , 8^k и т.д.) всегда записывается в двоичной системе как единица с соответствующим количеством нулей. Разность $2^k - 2^n$ при $k > n$ выглядит в двоичной системе как $k-n$ единиц с последующими n нулями. Этот факт — наглядный результат выполнения в двоич-

ной системе алгоритма вычитания «в столбик» (с заёмом единицы в старшем разряде). Убедиться в справедливости этого правила можно, записав в двоичной системе значение разности

$$128-32 (2^7-2^5) = 96 = 64+32 = 1100000_2$$

Таким образом, число 4^{2015} записывается в двоичной системе как единица с последующими 4030 нулями. Разность $4^{2015} - 2^{2014}$ выглядит как 2016 единиц и последующие 2014 нулей, а прибавление числа 3 (11_2) к этой разности заменяет два последних нуля единицами (все по алгоритму сложения в столбик). Итак, в двоичной записи этого выражения 2018 единиц.

Ответ: 2018.

Задачи для самостоятельного решения

- 16.1** Запись числа 53_{10} в системе счисления с основанием N оканчивается на 1 и содержит 3 цифры. Чему равно основание этой системы счисления?
- 16.2** Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 33_{10} оканчивается на 3.
- 16.3** В системе счисления с некоторым основанием число 56_{10} записывается в виде 211. Укажите это основание.
- 16.4** Укажите через запятую в порядке возрастания все числа, не превосходящие 30, запись которых в двоичной системе счисления оканчивается на 110. Числа в ответе указывать в десятичной системе счисления.
- 16.5** Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 26_{10} оканчивается на 1.
- 16.6** В системе счисления с некоторым основанием число 21_{10} записывается в виде 111. Укажите это основание.
- 16.7** Сколько единиц содержится в двоичной записи значения выражения:
 $4^{2013} + 2^{2012} - 17?$

1.3. Основы логики

Для решения задач по этой теме необходимо помнить таблицы истинности простейших функций, а также знать и уметь применять законы алгебры логики.

1) Таблицы истинности:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

2) Законы алгебры логики

Подобно алгебре чисел, алгебра логики имеет свои законы, записываемые формулами. Эти законы выражают свойства логических операций и используются при вычислении значений логических выражений.

Переместительный, распределительный и сочетательный законы справедливы для операций логического сложения и умножения.

Переместительный закон:

$$A \vee B = B \vee A$$

$$A \wedge B = B \wedge A$$

Сочетательный закон:

$$(A \vee B) \vee C = A \vee (B \vee C)$$

$$(A \wedge B) \wedge C = A \wedge (B \wedge C)$$

Распределительный закон:

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$$

Закон непротиворечия:

$$A \wedge \neg A = 0$$

Этот закон выражает тот факт, что высказывание не может быть одновременно истинным и ложным.

Закон исключенного третьего:

$$A \vee \neg A = 1$$

Этот закон означает, что либо высказывание, либо его отрицание должно быть истинным.

Закон двойного отрицания:

$$\neg(\neg A) = A$$

Законы де Моргана:

$$\neg(A \vee B) = \neg A \wedge \neg B$$

$$\neg(A \wedge B) = \neg A \vee \neg B$$

В справедливости указанных законов можно убедиться с помощью таблиц истинности.

Полезно знать также формулу для выражения импликации через отрицание и логическое сложение:

$$A \rightarrow B = \neg A \vee B$$

3) Приоритеты логических операций

В алгебре логики, так же как и в алгебре чисел, существует приоритет операций. От более старшей к младшей: $\neg \wedge \vee \rightarrow$, то есть, если не определено скобками, то сначала выполняется отрицание, потом умножение и т.д.

В варианте КИМ, которые получает каждый участник ЕГЭ, есть справочная страница (она идет второй, после страницы с инструкцией по выполнению варианта). На ней даются обозначения логических операций, содержатся сведения о приоритете операций и много другой полезной информации. Приведем здесь извлечение с этой страницы, касающееся рассматриваемого нами раздела:

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

а) *отрицание* (инверсия, логическое НЕ) обозначается \neg (например, $\neg A$);

б) *конъюнкция* (логическое умножение, логическое И) обозначается \wedge (например, $A \wedge B$) либо $\&$ (например, $A \& B$);

в) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается \vee (например, $A \vee B$);

г) *следование* (импликация) обозначается \rightarrow (например, $A \rightarrow B$);

е) *тождество* обозначается \equiv (например, $A \equiv B$). Выражение $A \equiv B$ истинно тогда и только тогда, когда значения A и B совпадают (либо они оба истинны, либо они оба ложны);

ф) символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

2. Два логических выражения, содержащих переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения $A \rightarrow B$ и $(\neg A) \vee B$ равносильны, а $A \vee B$ и $A \wedge B$ – нет (значения выражений разные, например, при $A = 1, B = 0$).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом, $\neg A \wedge B \vee C \wedge D$ означает то же, что и $((\neg A) \wedge B) \vee (C \wedge D)$.

Возможна запись $A \wedge B \wedge C$ вместо $(A \wedge B) \wedge C$. То же относится и к дизъюнкции: возможна запись $A \vee B \vee C$ вместо $(A \vee B) \vee C$.

Задание 2 – таблицы истинности

Пример:

Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z.

Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1

Какое выражение соответствует F?

- 1) $\neg X \wedge \neg Y \wedge Z$ 2) $\neg X \vee \neg Y \vee Z$ 3) $X \vee Y \vee \neg Z$ 4) $X \vee Y \vee Z$

Решение:

Чтобы не строить таблицу истинности для каждого выражения, можно просто перепроверить предложенные ответы. Для каждого из четырех вариантов нужно подставить значения X, Y, Z, приведенные в таблице, и сравнить полученные результаты со значением F, соответствующем данной строке:

1) $\neg X \wedge \neg Y \wedge Z = 0$ при $X = 0, Y = 0, Z = 0$, что не соответствует первой строке таблицы, где $F = 1$. Следовательно, данная функция нам не подходит;

2) $\neg X \vee \neg Y \vee Z = 1$ при $X = 0, Y = 0, Z = 1$, что не соответствует второй строке таблицы, где $F = 0$;

3) выражение $X \vee Y \vee \neg Z$ **соответствует F**, при всех предложенных комбинациях X, Y и Z;

Убедимся, что последний вариант не подходит:

4) $X \vee Y \vee Z = 1$ при $X = 0, Y = 0, Z = 1$, что не соответствует второй строке таблицы, где $F = 0$.

Ответ: 3

Задачи для самостоятельного решения

2.1

Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z.

Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	1	0
0	1	1	1
1	1	0	1

Какое выражение соответствует F?

- 1) $X \vee Y \vee \neg Z$ 2) $\neg X \vee Y \vee \neg Z$ 3) $\neg X \vee \neg Y \vee Z$ 4) $X \vee \neg Y \vee \neg Z$

2.2

Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z.

Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	0	0
1	0	1	1
1	1	0	0

Какое выражение соответствует F?

- 1) $\neg X \wedge \neg Y \wedge \neg Z$ 2) $X \wedge \neg Y \wedge Z$ 3) $X \wedge Y \wedge \neg Z$ 4) $\neg X \vee \neg Y \vee Z$

2.3

Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z.

Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	1	0	1
0	0	1	1
1	1	1	1

Какое выражение соответствует F?

- 1) $\neg X \wedge Y \wedge \neg Z$ 2) $X \vee \neg Y \vee Z$ 3) $X \vee Y \vee \neg Z$ 4) $\neg X \vee Y \vee Z$

2.4

Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z.

Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	1	0
0	1	0	0
1	1	0	0

Какое выражение соответствует F?

- 1) $X \vee Y \vee \neg Z$ 2) $X \wedge \neg Y \wedge Z$ 3) $X \wedge Y \wedge \neg Z$ 4) $\neg X \wedge Y \wedge \neg Z$

2.5

Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z.

Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	0	1
0	0	1	0
1	1	0	0

Какое выражение соответствует F?

- 1) $\neg X \wedge \neg Y \vee Z$ 2) $X \vee Y \vee \neg Z$ 3) $\neg X \wedge \neg Y \wedge \neg Z$ 4) $X \wedge Y \wedge Z$

2.6

Дан фрагмент таблицы истинности выражения F.

x1	x2	x3	x4	x5	x6	x7	x8	F
0	1	0	1	1	1	1	0	0
1	0	1	0	1	1	0	1	1
0	1	0	1	1	0	1	0	0

Каким выражением может быть F?

- 1) $x1 \wedge \neg x2 \wedge x3 \wedge \neg x4 \wedge x5 \wedge x6 \wedge \neg x7 \wedge x8$
 2) $\neg x1 \vee x2 \vee \neg x3 \vee x4 \vee \neg x5 \vee \neg x6 \vee x7 \vee \neg x8$
 3) $\neg x1 \wedge x2 \wedge \neg x3 \wedge x4 \wedge x5 \wedge x6 \wedge x7 \wedge \neg x8$
 4) $x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee \neg x6 \vee \neg x7 \vee x8$

2.7

Дан фрагмент таблицы истинности выражения F.

x1	x2	x3	x4	x5	x6	x7	x8	F
0	0	0	1	1	1	1	1	0
1	0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	1	1

Каким выражением может быть F?

- 1) $x1 \wedge \neg x2 \wedge x3 \wedge \neg x4 \wedge x5 \wedge x6 \wedge \neg x7 \wedge x8$
 2) $x1 \vee x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee \neg x6 \vee \neg x7 \vee \neg x8$
 3) $x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee x6 \vee \neg x7 \vee \neg x8$
 4) $\neg x1 \wedge x2 \wedge \neg x3 \wedge x4 \wedge x5 \wedge \neg x6 \wedge x7 \wedge x8$

Истинность логического выражения, содержащего импликацию

18 (A10 демоверсии 2012 г.)

Какое из приведенных имён удовлетворяет логическому условию:

(первая буква согласная \rightarrow вторая буква согласная) \wedge (предпоследняя буква гласная \rightarrow последняя буква гласная)

- 1) КРИСТИНА 2) МАКСИМ 3) СТЕПАН 4) МАРИЯ

Решение:

Поскольку логическое умножение истинно только тогда, когда обе его части истинны, то отсюда следует, что высказывание (первая буква согласная \rightarrow вторая буква согласная) истинно, и (предпоследняя буква гласная \rightarrow последняя буква гласная) истинно.

Заметим, что легче найти варианты, в которых импликации ложны и отсеять неправильные варианты.

Первая импликация ложна, когда высказывание (**первая буква согласная**) истинно, а (**вторая буква согласная**) – ложно. То есть, первая буква – согласная, а вторая – гласная. Следовательно, мы можем отбросить варианты МАКСИМ (2) и МАРИЯ (4).

Вторая импликация ложна, когда высказывание (**предпоследняя буква гласная**) истинно, а (**последняя буква гласная**) – ложно. То есть, предпоследняя буква имени – гласная, а последняя – согласная. Следовательно, мы можем отбросить вариант СТЕПАН (3).

Остается только имя КРИСТИНА, которое и является верным.

Ответ:1

18 (A10 демоверсии 2013 г.)

На числовой прямой даны два отрезка: $P = [2, 10]$ и $Q = [6, 14]$.

Выберите такой отрезок A , что формула

$$((x \in A) \rightarrow (x \in P)) \vee (x \in Q)$$

тождественно истинна, то есть принимает значение 1 при любом значении переменной x .

- 1) $[0, 3]$ 2) $[3, 11]$ 3) $[11, 15]$ 4) $[15, 17]$

Решение:

Выражение в скобках связывается с условием принадлежности x отрезку Q операцией «или». Это значит, что импликация должна быть истинна для всех точек x , которые не принадлежат данному отрезку. Импликация ложна в том случае, когда значение первого выражения $(x \in A)$ истинно, а второго $(x \in P)$ – ложно. Отрезок P частично перекрывается отрезком Q , поэтому значение всей дизъюнкции будет ложно для всех $x < 2$ и $x > 14$. Из всех предложенных в ответах отрезков только отрезок $[3, 11]$ лежит целиком внутри отрезка $[2, 14]$

Ответ:2

18 (A10 демоверсии 2014 г.)

На числовой прямой даны два отрезка: $P = [1, 39]$ и $Q = [23, 58]$.

Выберите из предложенных отрезков такой отрезок A , что логическое выражение

$$((x \in P) \rightarrow \neg(x \in Q)) \rightarrow \neg(x \in A)$$

тождественно истинно, то есть принимает значение 1 при любом значении переменной x .

- 1) $[5, 20]$ 2) $[25, 35]$ 3) $[40, 55]$ 4) $[20, 40]$

Решение:

Это задание решается достаточно просто: требуется применить преобразование импликации два раза и получить выражение без импликации:

$$\begin{aligned} ((x \in P) \rightarrow \neg(x \in Q)) \rightarrow \neg(x \in A) &\equiv (\neg(x \in P) \vee \neg(x \in Q)) \rightarrow \neg(x \in A) \equiv \\ &\neg(\neg(x \in P) \vee \neg(x \in Q)) \vee \neg(x \in A) \equiv (x \in P) \wedge (x \in Q) \vee \neg(x \in A) \end{aligned}$$

Понятно, что выражение будет тождественно истинно при любом значении переменной x только в том случае, если отрезок A будет целиком лежать внутри пересечения отрезков P и Q . Определив это пересечение: $P \cap Q = [23, 39]$, находим среди вариантов ответа отрезок, целиком лежащий внутри него: $[25, 35]$.

Ответ:2

Задачи для самостоятельного решения

18.1

Какое из приведенных имен удовлетворяет логическому условию:

вторая буква согласная \wedge \neg (последняя буква гласная \rightarrow первая буква согласная)

- 1) ГАЛИНА 2) СТАНИСЛАВ 3) АРТУР 4) АЛЕНА

18.2

Для какого слова истинно высказывание

(четвертая буква слова гласная \vee предпоследняя буква слова согласная) \rightarrow первая буква слова гласная

- 1) АТЛАС 2) БУЛКА 3) ЗАБОТА 4) КОРТ

18.3

Какое из приведенных имён удовлетворяет логическому условию:

(последняя буква согласная \rightarrow вторая буква согласная) \wedge \neg (третья буква гласная)

- 1) АЛЕКСЕЙ 2) ЕВГЕНИЯ 3) АРИНА 4) ПЕТР

18.4

Какое из приведенных имён удовлетворяет логическому условию:

(предпоследняя буква согласная \vee вторая буква согласная) \wedge (последняя буква гласная \rightarrow первая буква гласная)

- 1) СТЕПАН 2) КСЕНИЯ 3) ФЕДОР 4) НАТАЛЬЯ

18.5

Какое из приведенных имён удовлетворяет логическому условию:

(первая буква согласная \vee \neg (вторая буква согласная)) \wedge предпоследняя буква согласная

- 1) ПЕТР 2) БОРИС 3) МАРИЯ 4) ОЛЬГА

18.6

На числовой прямой даны два отрезка: $P = [2, 42]$ и $Q = [22, 62]$.

Выберите из предложенных отрезков такой отрезок A , что логическое выражение

$$((x \in P) \rightarrow \neg(x \in Q)) \rightarrow \neg(x \in A)$$

тождественно истинно, то есть принимает значение 1 при любом значении переменной x .

- 1) [3, 14] 2) [23, 32] 3) [43, 54] 4) [15, 45]

18.7

На числовой прямой даны два отрезка: $P = [1, 39]$ и $Q = [23, 58]$.

Выберите из предложенных отрезков такой отрезок A , что логическое выражение

$$((x \in P) \rightarrow (x \in Q)) \rightarrow \neg(x \in A)$$

тождественно истинно, то есть принимает значение 1 при любом значении переменной x .

- 1) [5, 20] 2) [25, 35] 3) [40, 55] 4) [20, 40]

18.8

На числовой прямой даны два отрезка: $P = [37; 60]$ и $Q = [40; 77]$. Укажите наименьшую возможную длину такого отрезка A , что формула

$$(x \in P) \rightarrow (((x \in Q) \wedge \neg(x \in A)) \rightarrow \neg(x \in P))$$

истинна при любом значении переменной x , то есть принимает значение 1 при любом значении переменной x .

Системы логических уравнений

23 (В15 демоверсии 2012 г.)

Сколько существует различных наборов значений логических переменных x_1, x_2, \dots, x_{10} , которые удовлетворяют всем перечисленным ниже условиям? $x_1 \equiv x_2$

$$((x_1 \equiv x_2) \vee (x_3 \equiv x_4)) \wedge (\neg(x_1 \equiv x_2) \vee \neg(x_3 \equiv x_4)) = 1$$

$$((x_3 \equiv x_4) \vee (x_5 \equiv x_6)) \wedge (\neg(x_3 \equiv x_4) \vee \neg(x_5 \equiv x_6)) = 1$$

...

$$((x_7 \equiv x_8) \vee (x_9 \equiv x_{10})) \wedge (\neg(x_7 \equiv x_8) \vee \neg(x_9 \equiv x_{10})) = 1$$

В ответе **не нужно** перечислять все различные наборы значений x_1, x_2, \dots, x_{10} , при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Решение:

В первую очередь найдем количество наборов x_1, x_2, x_3, x_4 , для которых выполняется первое равенство. Очевидно, что равенство верно только тогда, когда одновременно истинны два выражения $((x_1 \equiv x_2) \vee (x_3 \equiv x_4))$ и $(\neg(x_1 \equiv x_2) \vee \neg(x_3 \equiv x_4))$.

Истинность первого выражения достигается только тогда, когда хотя бы одна из двух пар x_1, x_2 и x_3, x_4 содержит эквивалентные между собой переменные. Второе выражение наоборот, будет истинным только в том случае, когда хотя бы одна из двух пар x_1, x_2 и x_3, x_4 содержит неэквивалентные между собой переменные.

Отсюда следует, что либо $(x_1 \equiv x_2)$ и $(x_3 \neq x_4)$, либо $(x_1 \neq x_2)$ и $(x_3 \equiv x_4)$. Таких наборов $4 \cdot 2 = 8$ (Для любой пары значений переменных x_1, x_2 подходят только 2 пары значений переменных x_3, x_4).

Во втором равенстве $((x_3 \equiv x_4) \vee (x_5 \equiv x_6)) \wedge (\neg(x_3 \equiv x_4) \vee \neg(x_5 \equiv x_6)) = 1$ добавляются переменные x_5, x_6 . Рассуждая так же, как и раньше, получим, что либо $(x_3 \equiv x_4)$ и $(x_5 \equiv x_6)$, либо $(x_3 \equiv x_4)$ и $(x_5 \neq x_6)$. Следовательно, если определено значение переменных x_3, x_4 , то существует 2 подходящих варианта значений переменных x_5, x_6 . Отсюда находим, что первым двум равенствам удовлетворяет $8 \cdot 2 = 16$ различных наборов переменных x_1, x_2, \dots, x_6 .

Рассуждая аналогично, получим $16 \cdot 2 = 32$ наборов переменных x_1, x_2, \dots, x_8 , удовлетворяющих первым трем равенствам и $32 \cdot 2 = 64$ набора переменных x_1, x_2, \dots, x_{10} , при которых выполнены все четыре равенства.

Ответ: 64

Задачи для самостоятельного решения

23.1

Сколько существует различных наборов значений логических переменных x_1, x_2, \dots, x_9 , которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \equiv x_2) \vee (x_1 \equiv x_3) \vee \neg(x_2 \equiv x_3) = 0$$

$$(x_3 \equiv x_4) \vee (x_3 \equiv x_5) \vee \neg(x_4 \equiv x_5) = 0$$

...

$$(x_7 \equiv x_8) \vee (x_7 \equiv x_9) \vee \neg(x_8 \equiv x_9) = 0$$

В ответе **не нужно** перечислять все различные наборы значений x_1, x_2, \dots, x_9 , при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

23.2

Сколько существует различных наборов значений логических переменных x_1, x_2, \dots, x_{10} , которые удовлетворяют всем перечисленным ниже условиям?

$$((x_1 \equiv x_3) \vee (x_2 \equiv x_4)) \wedge (\neg(x_1 \equiv x_3) \vee \neg(x_2 \equiv x_4)) = 1$$

$$((x_2 \equiv x_4) \vee (x_3 \equiv x_5)) \wedge (\neg(x_2 \equiv x_4) \vee \neg(x_3 \equiv x_5)) = 1$$

...

$$((x_7 \equiv x_9) \vee (x_8 \equiv x_{10})) \wedge (\neg(x_7 \equiv x_9) \vee \neg(x_8 \equiv x_{10})) = 1$$

В ответе **не нужно** перечислять все различные наборы значений x_1, x_2, \dots, x_{10} , при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

23.3

Сколько существует различных наборов значений логических переменных x_1, x_2, \dots, x_{10} , которые удовлетворяют всем перечисленным ниже условиям?

$$((x_1 \equiv x_2) \wedge (x_3 \equiv x_4)) \vee (\neg(x_1 \equiv x_2) \wedge \neg(x_3 \equiv x_4)) = 0$$

$$((x_3 \equiv x_4) \wedge (x_5 \equiv x_6)) \vee (\neg(x_3 \equiv x_4) \wedge \neg(x_5 \equiv x_6)) = 0$$

$$((x_5 \equiv x_6) \wedge (x_7 \equiv x_8)) \vee (\neg(x_5 \equiv x_6) \wedge \neg(x_7 \equiv x_8)) = 0$$

$$((x_7 \equiv x_8) \wedge (x_9 \equiv x_{10})) \vee (\neg(x_7 \equiv x_8) \wedge \neg(x_9 \equiv x_{10})) = 0$$

В ответе **не нужно** перечислять все различные наборы значений переменных x_1, x_2, \dots, x_{10} , при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

1.4. Моделирование

В задачах по этому разделу курса обычно требуется установить соответствие между разными видами представления данных, чаще всего – между таблицей и графом. При решении задач на эту тему необходимо уметь преобразовывать табличные модели (таблицы) в эквивалентные сетевые модели, представленные в виде схем. И, наоборот, по заданной схеме уметь строить эквивалентную табличную модель.

5 (А2 демоверсии 2012 г.)

Между населёнными пунктами А, В, С, D, E, F построены дороги, протяженность которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

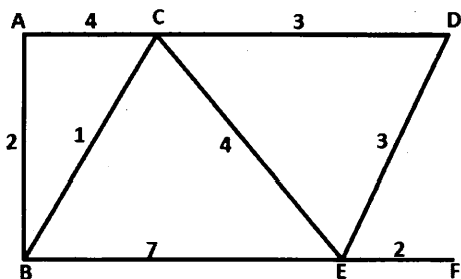
	A	B	C	D	E	F
A		2	4			
B	2		1		7	
C	4	1		3	4	
D			3		3	
E		7	4	3		2
F					2	

Определите длину кратчайшего пути между пунктами А и F (при условии, что передвигаться можно только по построенным дорогам).

- 1) 9 2) 10 3) 11 4) 12

Решение:

Нарисуем схему, соответствующую таблице:



Рассмотрим возможные пути между конечными пунктами и их длины:

$$A \rightarrow B \rightarrow E \rightarrow F = 11;$$

$$A \rightarrow C \rightarrow E \rightarrow F = 10;$$

$$A \rightarrow C \rightarrow D \rightarrow E \rightarrow F = 12;$$

$$A \rightarrow B \rightarrow C \rightarrow E \rightarrow F = 9.$$

Остальные пути можно не рассматривать, т.к. их длина значительно больше, чем 9.

Следовательно кратчайший путь равен 9 и пролегает по вершинам в таком порядке:

$$A \rightarrow B \rightarrow C \rightarrow E \rightarrow F.$$

Ответ: 1

Задачи для самостоятельного решения

5.1

Между населёнными пунктами A, B, C, D, E, F построены дороги, протяженность которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		5	4			3
B	5		2			4
C	4	2		12		
D			12		4	
E				4		2
F	3	4			2	

Определите длину кратчайшего пути между пунктами C и D (при условии, что передвигаться можно только по построенным дорогам).

- 1) 10 2) 11 3) 12 4) 13

5.2

Между населёнными пунктами А, В, С, D, E, F построены дороги, протяженность которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		4		10	1	
B	4		4	8		1
C		4		5		
D	10	8	5			
E	1					1
F		1			1	

Определите длину кратчайшего пути между пунктами А и D (при условии, что передвигаться можно только по построенным дорогам).

- 1) 10 2) 11 3) 12 4) 9

5.3

Между населёнными пунктами А, В, С, D, E, F построены дороги, протяженность которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A			5			
B			2	2		
C	5	2			7	3
D		2			1	
E			7	1		2
F			3		2	

Определите длину кратчайшего пути между пунктами А и E (при условии, что передвигаться можно только по построенным дорогам).

- 1) 10 2) 9 3) 8 4) 7

5.4

Между населёнными пунктами А, В, С, D, E, F построены дороги, протяженность которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		2		6		5
B	2		2	1		
C		2			5	
D	6	1			2	
E			5	2		5
F	5				5	

Определите длину кратчайшего пути между пунктами А и E (при условии, что передвигаться можно только по построенным дорогам).

- 1) 6 2) 5 3) 8 4) 7

5.5

Между населёнными пунктами А, В, С, D, E, F построены дороги, протяженность которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

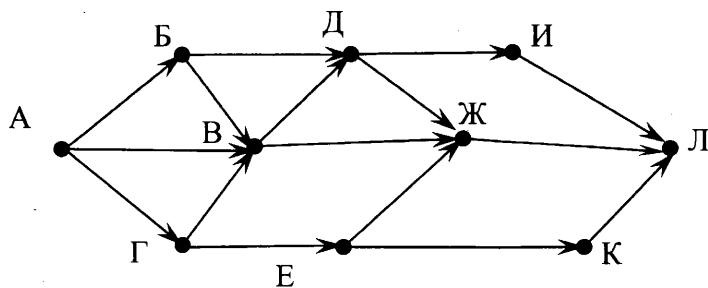
	A	B	C	D	E	F
A		4				3
B	4		3		4	4
C		3		3	2	
D			3		4	
E		4	2	4		
F	3	4				

Определите длину кратчайшего пути между пунктами D и F (при условии, что передвигаться можно только по построенным дорогам).

- 1) 12 2) 10 3) 13 4) 9

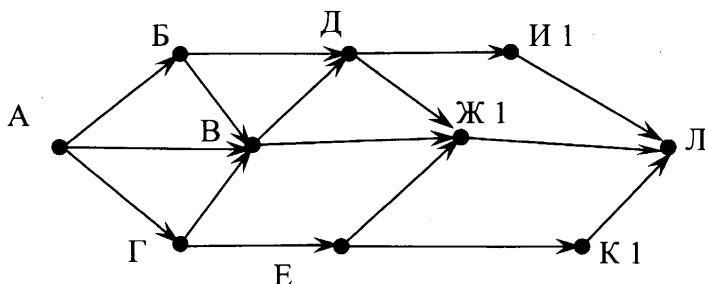
15 (В9 демоверсии 2013 г.)

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Л?

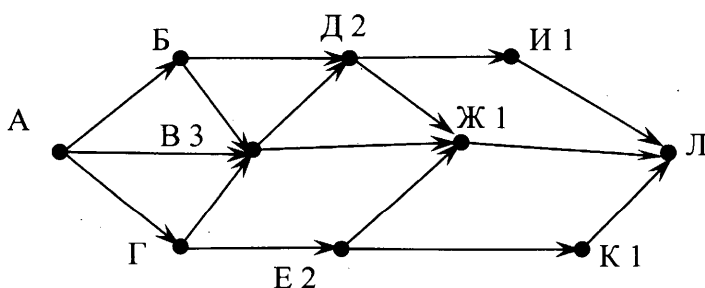


Решение:

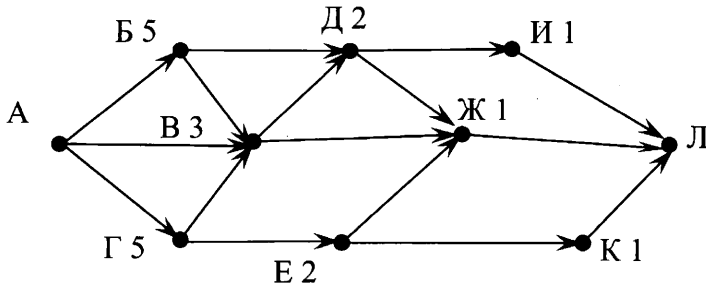
Это задача так называемого динамического программирования. Ключевая идея в динамическом программировании достаточно проста. Как правило, чтобы решить поставленную задачу, требуется решить отдельные части задачи (подзадачи), после чего объединить решения подзадач в одно общее решение. Часто многие из этих подзадач одинаковы. Подход динамического программирования состоит в том, чтобы решить каждую подзадачу только один раз, сократив тем самым количество вычислений. Это особенно полезно в случаях, когда число повторяющихся подзадач велико. В данном случае начнем решать задачу с конца – от города Л. Видно, что от городов И, Ж и К в Л можно попасть единственным путем. Напишем единичку возле названий этих городов:



Из города Д в направлении Л выходят две дороги – одна в И, другая в Ж. Из указанных городов путь в Л, как мы помним, единственный. Аналогично из Е в Л можно проехать тоже двумя путями - через Ж и через К. В городе В путник может отправиться в Л напрямую через Ж (один путь) или через Д (два пути). Таким образом, общее количество путей из В в Л – три (2+1). Нарисуем это на схеме:



Продолжая этот метод, мы видим, что из Б в Л существуют 5 путей: два через Д и три через В. Аналогично для Г: 3 пути через В и 2 пути через Е, в сумме 5.



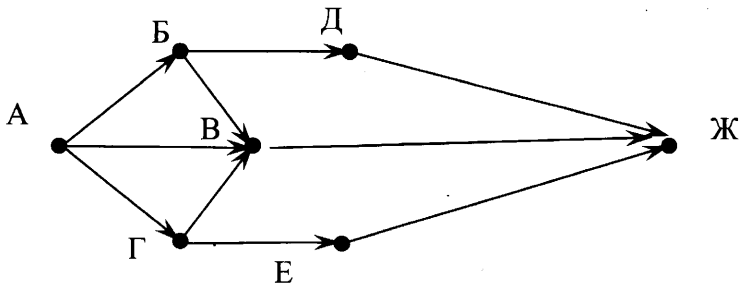
Таким образом, из А в Л можно попасть 13 путями: 5 через Б, 3 через В и 5 через Г.

Ответ: 13

Задачи для самостоятельного решения

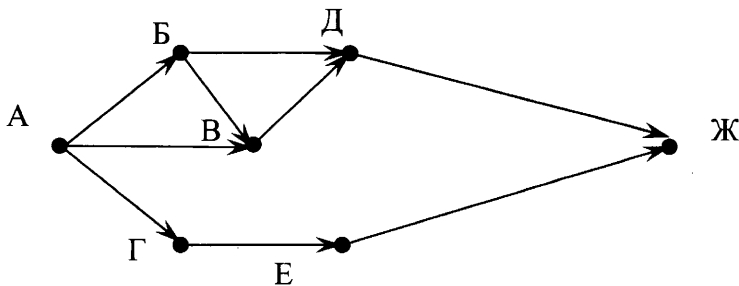
15.1

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж?



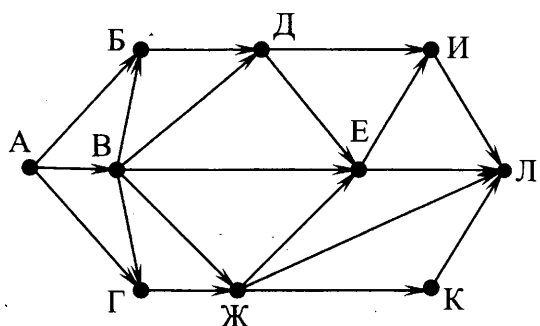
15.2

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж?



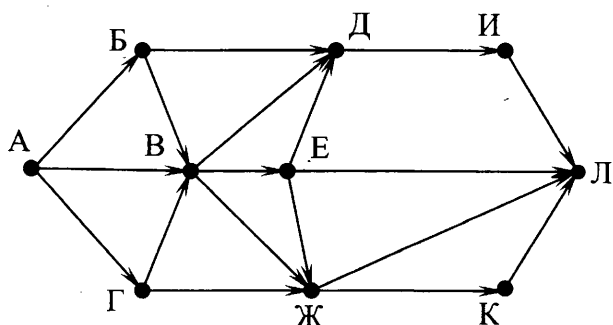
15.3

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Л?



15.4

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Л?



2. Алгоритмизация и программирование

2.1. Исполнение алгоритмов

В этом разделе рассматриваются задания ЕГЭ связанные как с непосредственным исполнением, так и с конструированием и анализом алгоритмов для различных исполнителей. Во многих заданиях ЕГЭ по этой теме в ходе решения необходимо провести анализ алгоритма, но вовсе не требуется его пошагового выполнения.

6 (А5 демоверсии 2012 г.)

Автомат получает на вход два трехзначных числа. По этим числам строится новое число по следующим правилам.

Вычисляются три числа – сумма старших разрядов заданных трехзначных чисел, сумма средних разрядов этих чисел, сумма младших разрядов.

Полученные три числа записываются друг за другом в порядке убывания (без разделителей).

Пример. Исходные трехзначные числа: 835, 196. Поразрядные суммы: 9, 12, 11. Результат: 12119

Определите, какое из следующих чисел может быть результатом работы автомата.

- 1) 151303 2) 161410 3) 191615 4) 121613

Решение:

В задаче не требуется исполнение алгоритма, но необходимо сформулировать свойства возможного результата его работы, для того, чтобы выделить тот вариант ответа, который может являться результатом алгоритма.

В соответствии с условием, свойства допустимого результата следующие:

1. он состоит не менее, чем из трех, и не более, чем из шести цифр
2. его можно разбить ровно на три последовательно записанных числа, каждое из которых состоит из одной или двух цифр
 - 2.1. каждое из этих трех чисел больше или равно нулю (0+0), и меньше или равно 18 (9+9).
 - 2.2. двузначное число не может иметь ведущего нуля, т.е. запись типа 05 недопустима
 - 2.3. эти три двузначных числа записаны слева направо, в порядке убывания.

Рассмотрим поочередно все варианты ответов:

- 1) 151303 можно разбить на 3 группы, соблюдая условие 2, единственным образом: 15 13 03. При этом нарушается условие 2.2. Следовательно, этот вариант ответа не подходит.
- 2) 161410 тоже разбивается единственным образом: 16 14 10 – все условия выполняются, **вариант подходит.**

Убедимся, что остальные варианты ответов не подходят

- 3) 191615 разбивается 19 16 15 – не подходит, т.к. 19 недопустимо (п. 2.1)
- 4) 121613 разбивается 12 16 13 – не подходит, т.к. числа неупорядочены.

Ответ: 2

Задачи для самостоятельного решения

6.1

Предлагается некоторая операция над двумя произвольными трёхзначными десятичными числами:

1. Записывается результат сложения значений старших разрядов заданных чисел.
 2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому слева, иначе – справа.
 3. Итоговое число получают приписыванием справа к полученному после второго шага числу суммы значений младших разрядов исходных чисел.
- Определите, какое из предложенных чисел может быть результатом такой операции.

- 1) 131615
- 2) 151316
- 3) 151620
- 4) 131605

6.2

Условие такое же, как в 6.1

Определите, какое из предложенных чисел может быть результатом операции.

- 1) 131703
- 2) 151710
- 3) 171513
- 4) 191715

6.3

Условие такое же, как в 6.1

Определите, какое из предложенных чисел может быть результатом операции.

- 1) 131712
- 2) 131214
- 3) 121407
- 4) 172114

6.4

Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв – из двух бит, для некоторых – из трех). Эти коды представлены в таблице:

A	B	C	D	E
000	11	01	001	10

Из четырех полученных сообщений в этой кодировке только одно прошло без ошибки и может быть корректно декодировано. Найдите его.

- 1) 110001001001110
- 2) 110000011011110
- 3) 110000010011110
- 4) 110000001011110

6.5

Цепочка из трех бусин, помеченных латинскими буквами, формируется по следующему правилу. В конце цепочки стоит одна из бусин А, В, С. На первом месте – одна из бусин В, D, С, которой нет на третьем месте. В середине – одна из бусин А, С, Е, В, не стоящая на первом месте. Какая из перечисленных цепочек создана по этому правилу?

- 1) СВВ 2) ЕАС 3) ВСD 4) ВСВ

6.6

Саша и Женя играют в такую игру. Саша пишет слово русского языка. Женя заменяет в нем каждую букву на другую букву так, чтобы были выполнены такие правила.

- 1) Гласная буква меняется на согласную, согласная – на гласную.
- 2) Количество букв в слове не изменяется.
- 3) В получившемся слове буквы следуют в алфавитном порядке.

Пример. Саша написала: ЖЕНЯ. Женя может написать, например, ЕНОТ или ЁЖИК.

Но не может написать МАМА или ИВАН.

Для справки. В алфавите буквы идут в таком порядке:

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

Саша написала: КОТ.

Укажите, какое из следующих слов может написать Женя

- 1) АНЯ 2) ЁЛКА 3) ЖИР 4) ЭЛЯ

6.7

В некоторой информационной системе информация кодируется двоичными шестиразрядными словами. При передаче данных возможны их искажения, поэтому в конец каждого слова добавляется седьмой (контрольный) разряд таким образом, чтобы сумма разрядов нового слова, считая контрольный, была чётной. Например, к слову 110011 справа будет добавлен 0, а к слову 101100 – 1.

После приёма слова производится его обработка. При этом проверяется сумма его разрядов, включая контрольный. Если она нечётна, это означает, что при передаче этого слова произошёл сбой, и оно автоматически заменяется на зарезервированное слово 0000000. Если она чётна, это означает, что сбоя не было или сбоев было больше одного. В этом случае принятое слово не изменяется.

Исходное сообщение

1101001 0011000 0011101

было принято в виде

1101001 0011101 0011100.

Как будет выглядеть принятое сообщение после обработки?

- 1) 1101001 0000000 0011100
- 2) 0000000 0011101 0011100
- 3) 1101001 0011101 0000000
- 4) 1101001 0000000 0000000

6.8

В некоторой информационной системе информация кодируется двоичными шестиразрядными словами. При передаче данных возможны их искажения, поэтому в конец каждого слова добавляется седьмой (контрольный) разряд таким образом, чтобы сумма разрядов нового слова, считая контрольный, была чётной. Например, к слову 110011 справа будет добавлен 0, а к слову 101100 – 1.

После приёма слова производится его обработка. При этом проверяется сумма его разрядов, включая контрольный. Если она нечётна, это означает, что при передаче этого слова

произошёл сбой, и оно автоматически заменяется на зарезервированное слово 0000000. Если она чётна, это означает, что сбоя не было или сбоев было больше одного. В этом случае принятое слово не изменяется.

Исходное сообщение

1000100 1111101 1101001

было принято в виде

1000101 1111101 1110001.

Как будет выглядеть принятое сообщение после обработки?

- 1) 0000000 1111101 0000000
- 2) 0000000 1111101 1110001
- 3) 1000101 1111101 0000000
- 4) 1000100 0000000 1101001

В той же позиции экзаменационной работы 2015 года может стоять задача, где алгоритм записывается не на естественном языке, а на языке арифметического исполнителя с ограниченным набором команд. Задание обычно связано с нахождением алгоритма, содержащего наименьшее количество команд.

6 (В2 демоверсии 2012 г.)

У исполнителя Утроитель две команды, которым присвоены номера:

1. прибавь 1,

2. умножь на 3.

Первая из них увеличивает число на экране на 1, вторая – утраивает его.

Запишите порядок команд в программе преобразования числа 1 в число 22, содержащей не более 5 команд, указывая лишь номера команд. (Например, **21211** – это программа

умножь на 3

прибавь 1

умножь на 3

прибавь 1

прибавь 1,

которая преобразует число 1 в 14.)

(Если таких программ более одной, то запишите любую из них.)

Решение:

Число 22 Утроитель может получить единственным способом, прибавив единицу к 21. Поэтому последняя команда программы – 1. Число 21 Утроитель может получить двумя способами – утроив 7 или последовательно прибавив три единицы к 18. Поскольку наша программа может содержать не более 5 команд, то второй способ не подходит (оставшейся одной командой получить 18 из 1 нельзя). Итак, предпоследняя команда программы – 2, соответствующая получению 21 из 7. Число 7 Утроитель может получить единственным способом, прибавив единицу к 6, следовательно, третья от конца команда – 1. Учитывая, что число 6 получается утроением 2, а 2 – прибавлением 1 к 1, можно выписать получившуюся программу из 5 команд: 12121.

Выполним проверку

Исходное число	Команда	Результат
1	1 (прибавь 1)	2
2	2 (умножь на 3)	6
6	1 (прибавь 1)	7
7	2 (умножь на 3)	21
21	1 (прибавь 1)	22

Ответ: 12121

Задачи для самостоятельного решения (ОС)

6.9

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 2

2. умножь на 3

Выполняя первую из них, Калькулятор прибавляет к числу на экране 2, а выполняя вторую, утраивает его. Запишите порядок команд в программе получения из 0 числа 28, содержащей не более 6 команд, указывая лишь номера команд. (Например, программа **21211** – это программа:

умножь на 3

прибавь 2

умножь на 3

прибавь 2

прибавь 2,

которая преобразует число 1 в 19.)

6.10

У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат,

2. прибавь 1.

Первая из них возводит число на экране в квадрат, вторая – увеличивает его на 1. Запишите порядок команд в программе получения из числа 1 числа 10, содержащей не более 4 команд, указывая лишь номера команд. (Например, программа **12122** – это программа:

возведи в квадрат,

прибавь 1,

возведи в квадрат,

прибавь 1,

прибавь 1,

которая преобразует число 1 в 6.)

6.11

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 3

2. умножь на 2

Выполняя первую из них, Калькулятор прибавляет к числу на экране 3, а выполняя вторую, удваивает его. Запишите порядок команд в программе получения из 1 числа 23, содержащей не более 6 команд, указывая лишь номера команд. (Например, программа **21211** – это программа:

умножь на 2

прибавь 3

умножь на 2

прибавь 3

прибавь 3,

которая преобразует число 1 в 16.)

6.12

У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат,

2. прибавь 1.

Первая из них возводит число на экране в квадрат, вторая увеличивает его на 1. Запишите порядок команд в программе, которая преобразует число 3 в число 26 и содержит не более 4 команд. Указывайте лишь номера команд.

(Например, программа **2122** – это программа

прибавь 1,

возведи в квадрат,

прибавь 1,

прибавь 1.

Эта программа преобразует число 3 в число 18.)

6.13

У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат,
2. прибавь 1.

Первая из них возводит число на экране в квадрат, вторая увеличивает его на 1. Запишите порядок команд в программе, которая преобразует число 2 в число 36 и содержит не более 4 команд. Указывайте лишь номера команд.

(Например, программа 2122 – это программа

- прибавь 1,
возведи в квадрат,
прибавь 1,
прибавь 1.

Эта программа преобразует число 1 в число 6.)

Задача анализа алгоритма может предполагать как подсчет различных результатов исполнения алгоритма для различных исходных данных, так и подсчет количества различных алгоритмов приводящих исполнителя из исходного положения в искомое. Методы решения этих задач также могут существенно различаться. Решения также могут в некоторых случаях получаться методом устных рассуждений, а в некоторых случаях требуют кропотливых вычислений.

22 (В13 демоверсии 2012)

У исполнителя Кузнечик две команды:

1. прибавь 3,
2. вычти 2.

Первая из них увеличивает число на экране на 1, вторая – уменьшает его на 2 (отрицательные числа допускаются).

Программа для Кузнечика – это последовательность команд. Сколько различных чисел можно получить из числа 1 с помощью программы, которая содержит ровно 5 команд?

Решение:

В этой задаче также не требуется непосредственное исполнение алгоритма, но нужен его анализ. Заметим, что результат этого алгоритма зависит только от соотношения количества команд «прибавь 3» и «вычти 2». Так, например, если в программе три команды сложения и две команды вычитания, то результат будет один и тот же, независимо от порядка следования команд в программе. Поскольку количество команд в программе фиксировано, то можно считать, что результат однозначно определяется количеством команд одного вида, например, команд сложения (если в программе N_1 команд сложения, то команд вычитания не может быть иным, чем $N_2 = 5 - N_1$). Понятно, что в программе из пяти команд может присутствовать только 0, 1, 2, 3, 4 или 5 команд сложения, т.е. возможно всего шесть семейств программ, со своим результатом для каждого семейства.

Ответ: 6

22 (В13 демоверсии 2013 г.)

У исполнителя Удвоитель две команды, которым присвоены номера:

1. прибавь 1,
2. умножь на 2.

Первая из них увеличивает на 1 число на экране, вторая удваивает его.

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые число 3 преобразуют в число 23?

Решение:

Эту задачу можно решить с помощью следующих рассуждений:

Числа 4 и 5 можно получить из числа 3 с помощью Удвоителя единственным образом: прибавляя 1 к предыдущему числу. Число 6 двумя способами: прибавив 1 к числу 5 и удвоив число 3. Число 7 получается из числа 6 единственным способом: прибавлением еди-

ницы, а из числа 3 – двумя способами, так как число 6 можно получить также двумя способами. (Запишем для наглядности программы получения числа 7 из числа 3: **1111** и **21**.) Число 8 можно получить уже тремя способами: прибавив 1 к числу 7 или удвоив число 4, получаемое из 3 единственным образом – прибавлением единицы. Аналогично число 9 получается тем же числом способов, что и число 8 (трижды способами).

Обобщив эти наблюдения, сделаем вывод: четное число получается, начиная с числа 6, либо прибавлением единицы к предшествующему числу, либо удвоением числа вдвое меньшего. При этом количество способов получения этого числа суммируется. Следующее за ним нечетное число получается из предшествующего прибавлением единицы, поэтому количество способов получения у нечетного числа и предшествующего ему четного совпадают.

Построим таблицу, в которой в первом ряду напишем числа, а во втором ряду – количество способов их получения из числа 3 с помощью Удвоителя.

4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	1	2	2	3	3	4	4	6	6	8	8	11	11	14	14	18	18	22	22

Видно, что число 23 получается из числа 22 прибавлением единицы, а вот число 22 может быть получено 22 способами: прибавлением единицы к числу 21, получаемому 18 способами или удвоением числа 11, получаемого 4 способами. Таким образом, число 23 получается из числа 3 с помощью 22 различных программ для Удвоителя.

Следует обратить внимание, что эта задача никоим образом не предполагает выписывания всех этих 22 программ (и, тем более, доказывания, что все остальные программы не приводят к искомому результату).

Ответ: 22

Задачи для самостоятельного решения

22.1

У исполнителя Кузнечик две команды:

- 1. прибавь 6,**
- 2. вычти 5.**

Первая из них увеличивает число на экране на 6, вторая – уменьшает его на 5 (отрицательные числа допускаются).

Программа для Кузнечика – это последовательность команд. Сколько различных чисел можно получить из числа 1 с помощью программы, которая содержит ровно 7 команд?

22.2

У исполнителя Кузнечик две команды:

- 1. прибавь 6,**
- 2. вычти 5.**

Первая из них увеличивает число на экране на 6, вторая – уменьшает его на 5 (отрицательные числа не допускаются).

Программа для Кузнечика – это последовательность команд. Сколько различных чисел можно получить из числа 0 с помощью программы, которая содержит ровно 7 команд?

22.3

У исполнителя Кузнечик две команды:

- 1. прибавь 6,**
- 2. вычти 6.**

Первая из них увеличивает число на экране на 6, вторая – уменьшает его на 6 (отрицательные числа допускаются).

Программа для Кузнечика – это последовательность команд. Сколько различных чисел можно получить из числа 1 с помощью программы, которая содержит ровно 8 команд?

22.4

У исполнителя Кузнечик две команды:

1. прибавь 6,

2. вычти 6.

Первая из них увеличивает число на экране на 6, вторая – уменьшает его на 6 (отрицательные числа не допускаются).

Программа для Кузнечика – это последовательность команд. Сколько различных чисел можно получить из числа 3 с помощью программы, которая содержит ровно 8 команд?

В демоверсии 2012 года был представлен новый тип задач С3, при решении которых учащийся должен продемонстрировать умение проанализировать модель предложенного исполнителя алгоритмов, сформулировать выводы и строго обосновать их. В экзамене 2013 года авторы вернулись к задачам на поиск и обоснование оптимальной стратегии игры, а задания этого содержания (в упрощенном виде, не требовавшие обоснования, а только вычисления количества возможных программ) были даны в позиции В13. В 2015 году аналогичные задания будут стоять в позиции 22. Тем не менее мы считаем полезным привести здесь разбор задания С3 2012 года.

22 (С3 демоверсии 2012 г.)

У исполнителя Утроитель две команды, которым присвоены номера:

1. прибавь 1,

2. умножь на 3.

Первая из них увеличивает число на экране на 1, вторая – утраивает его.

Программа для Утроителя – это последовательность команд.

Сколько есть программ, которые число 1 преобразуют в число 29?

Ответ обоснуйте.

Решение:

Способ 1.

Обозначим $R(n)$ – количество программ, которые преобразуют число 1 в число n . Обозначим $t(n)$ наибольшее кратное трем, не превосходящее n .

Обе команды исполнителя увеличивают исходное число, поэтому общее количество команд в программе не может превосходить 28.

Верны следующие соотношения:

Если n не делится на 3, то тогда $R(n) = R(t(n))$, так как существует единственный способ получения n из $t(n)$ – прибавлением единиц.

Пусть n делится на 3. Тогда $R(n) = R(n/3) + R(n-1) = R(n/3) + R(n-3)$ (если $n > 3$). При $n=3$ $R(n) = 2$ (два способа: прибавлением двух единиц или однократным умножением на 3). Поэтому достаточно последовательно вычислить значения $R(n)$ для всех чисел, кратных трем и не превосходящих 29. Имеем:

$$R(2)=1$$

$$R(3) = 2 = R(4)=R(5)$$

$$R(6) = R(2)+R(3) = 1+2 = 3 = R(7)=R(8)$$

$$R(9) = R(3)+R(6) = 2+3 = 5 = R(10)=R(11)$$

$$R(12) = R(4)+R(9) = 2+5 = 7 = R(13)=R(14)$$

$$R(15) = R(5)+R(12) = 2+7 = 9 = R(16)=R(17)$$

$$R(18) = R(6)+R(15) = 3+9 = 12 = R(19)=R(20)$$

$$R(21) = R(7)+R(18) = 3+12 = 15 = R(22)=R(23)$$

$$R(24) = R(8)+R(21) = 3+15 = 18 = R(25)=R(26)$$

$$R(27) = R(9)+R(24) = 5+18 = 23 = R(28)=R(29)$$

Ответ: 23

Способ 2.

Будем решать поставленную задачу последовательно для чисел 1, 2, 3, ..., 29 (то есть для каждого из чисел определим, сколько программ исполнителя существует для его получения). Количество программ, которые преобразуют число 1 в число n , будем обозначать через $R(n)$. Число 1 у нас уже есть, значит, его можно получить с помощью “пустой” программы. Любая непустая программа увеличит исходное число, т.е. даст число, больше 1. Значит, $R(1) = 1$. Для каждого следующего числа рассмотрим, из какого числа оно может быть получено за одну команду исполнителя. Если число не делится на три, то оно может быть получено только из предыдущего с помощью команды **прибавь 1**. Значит, количество искомых программ для такого числа равно количеству программ для предыдущего числа: $R(i) = R(i-1)$. Если число на 3 делится, то вариантов последней команды два: **прибавь 1** и **умножь на 3**, тогда $R(i) = R(i-1) + R(i/3)$. Заполним соответствующую таблицу по приведенным формулам слева направо:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	2	2	3	3	3	5	5	5	7	7	7	9
16	17	18	19	20	21	22	23	24	25	26	27	28	29	
9	9	12	12	12	15	15	15	18	18	18	23	23	23	

При этом ячейки, относящиеся к числам, которые не делятся на 3, можно в решении и опустить (за исключением первого и последнего чисел):

1	3	6	9	12	15	18	21	24	27	28	29
1	2	3	5	7	9	12	15	18	23	23	23

Ответ: 23

Эту задачу можно решить, построив дерево (древовидный граф), дуги которого соответствуют операциям добавления единицы или утроения, а вершины – результатам вычислений. Поскольку дерево для приведенной выше задачи занимает на странице много места, для иллюстрации этого способа, приведем решение аналогичной задачи с более компактным деревом.

Пример:

У исполнителя Утроитель две команды, которым присвоены номера:

1. **прибавь 1,**

2. **умножь на 3.**

Первая из них увеличивает число на экране на 1, вторая – утраивает его.

Программа для Утроителя – это последовательность команд.

Сколько есть программ, которые число 3 преобразуют в число 16?

Ответ обоснуйте.

Решение:

Любое целое число большее 3 может быть получено данным исполнителем одним из двух способов:

1) добавлением единицы (так можно получить любое число)

2) умножением на 3 (так можно получить только число, кратное 3).

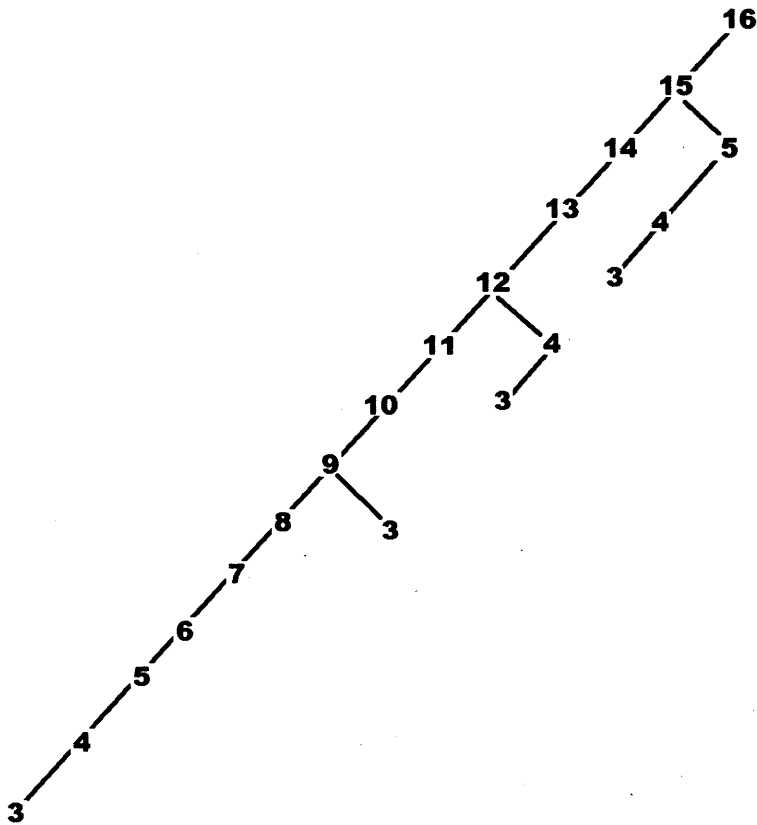
Построим бинарное (двоичное) дерево со следующими свойствами:

– в корне дерева поместим искомый результат – 16;

– левый потомок для каждой вершины либо отсутствует, либо вершина получается из него добавлением единицы;

– правый потомок для каждой вершины либо отсутствует, либо вершина получается из него умножением на 3;

Другими словами, левая ссылка соответствует вычитанию из значения вершины единицы, а правая – делению на 3 (возможно только для чисел, кратных 3 и больших 6).



Каждый путь от вершины со значением 3 к корню дерева соответствует одной программе Утроителя получения из числа 3 числа 16. Таких путей 4. Следовательно,

Ответ :4.

В качестве упражнения предлагается самостоятельно решить эту задачу двумя приведенными выше способами, а также построить двоичное дерево для задачи из демоверсии.

Задачи для самостоятельного решения

22.5

У исполнителя Удвоитель две команды, которым присвоены номера:

1. прибавь 1,
2. умножь на 2.

Первая из них увеличивает число на экране на 1, вторая удваивает его.

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 22?

22.6

У исполнителя Прибавитель две команды, которым присвоены номера:

1. прибавь 1,
2. увеличь старшую цифру числа на 1.

Первая из них увеличивает число на экране на 1, вторая увеличивает на 1 старшую (левую) цифру числа, например число 23 с помощью такой команды превратится в число 33. Если старшая цифра числа равна 9, то вторая команда оставляет это число неизменным.

Программа для Прибавителя – это последовательность команд.

Сколько есть программ, которые число 15 преобразуют в число 38?

22.7 (демоверсия 2015 г.)

Исполнитель Май4 преобразует число, записанное на экране. У исполнителя три команды, которым присвоены номера:

1. Прибавь 1
2. Прибавь 2
3. Прибавь 4

Первая из них увеличивает число на экране на 1, вторая увеличивает это число на 2, а третья – на 4. Программа для исполнителя Май4 – это последовательность команд.

Сколько есть программ, которые **число 21** преобразуют в **число 30**?

22.8

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 2
2. умножь на 3

Первая из них увеличивает число на экране на 2, вторая – утраивает его.

Программа для Калькулятора – это последовательность команд.

Сколько есть программ, которые **число 1** преобразуют в **число 31**?

22.9

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 3
2. умножь на 2

Первая из них увеличивает число на экране на 3, вторая – удваивает его.

Программа для Калькулятора – это последовательность команд.

Сколько есть программ, которые **число 1** преобразуют в **число 31**?

22.10

У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат,
2. прибавь 1.

Первая из них возводит число на экране в квадрат, вторая – увеличивает его на 1.

Программа для Квадратора – это последовательность команд.

Сколько есть программ, которые **число 2** преобразуют в **число 100**?

22.11

У исполнителя Удвоитель две команды, которым присвоены номера:

1. прибавь 3
2. умножь на 2

Первая из них увеличивает число на экране на 3, вторая – удваивает его.

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые **число 4** преобразуют в **число 43**?

22.12

У исполнителя Удвоитель две команды, которым присвоены номера:

1. прибавь 1
2. умножь на 2

Первая из них увеличивает число на экране на 1, вторая – удваивает его.

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые **число 1** преобразуют в **число 25**?

22.13

У исполнителя Удвоитель две команды, которым присвоены номера:

1. прибавь 2

2. умножь на 2

Первая из них увеличивает число на экране на 2, вторая – удваивает его.

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые **число 2** преобразуют в **число 28**?

22.14

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 3

2. умножь на 3

Первая из них увеличивает число на экране на 3, вторая – утраивает его.

Программа для Калькулятора – это последовательность команд.

Сколько есть программ, которые **число 3** преобразуют в **число 119**?

Ответ обоснуйте.

Задание 26 проверяет умение построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию. В демоверсии 2015 года задание 26 продолжает линию, начатую заданием С3 демоверсии 2013 года, которое синтезировало подходы 2012 года и предшествующих лет, когда требовалось найти выигрывающего игрока для игры с камнями. Сейчас анализируется игра с камнями и выигрышные стратегии для этой игры, но задача состоит в определении исходного положения, в котором у того или иного игрока существует однозначная выигрышная стратегия. Приводим здесь не только задания в нынешнем формате, но и задания прежних лет.

26 (С3 демоверсии 2013 г.)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** камень или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 22. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 22 и более камней.

В начальный момент в куче было S камней, $1 \leq S \leq 21$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрывающий ход для каждого указанного значения S .

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два таких значения S , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

Решение:

1. а) Петя может выиграть первым ходом, если $S = 11, \dots, 21$. Во всех этих случаях достаточно удвоить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 21 камня.

б) Ваня может выиграть своим первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 10$ камней. Тогда после первого хода Пети в куче будет 11 или 20 камней. В обоих случаях Ваня утраивает количество камней и выигрывает в один ход.

2. Возможные значения S : 5 и 9. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 10 камней: в первом случае удвоением, во втором добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.

3. Возможное значение S : 8. После первого хода Пети в куче будет 9 или 16 камней. Если в куче станет 16 камней, Ваня удвоит количество камней и выигрывает первым ходом. Ситуация, когда в куче 9 камней, уже разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты.

Положения после очередных ходов				
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
8	$8+1=9$	$9+1=10$	$10+1=11$	<u>$11 \cdot 2=22$</u>
			$10 \cdot 2=20$	<u>$20 \cdot 2=40$</u>
	$8 \cdot 2=16$	<u>$16 \cdot 2=32$</u>		

26 (демоверсия 2015 г.)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **три** камня или увеличить количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 18 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 35.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 35 или больше камней.

В начальный момент в куче было S камней; $1 \leq S \leq 34$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

Задание 1

а) Укажите все такие значения числа S , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрывающие ходы.

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

Задание 2

Укажите два таких значения S , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для каждого указанного значения S опишите выигрышную стратегию Пети.

Задание 3

Укажите значение S , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани.

Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рисунке на рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.

Решение:

Задание 1

а) Петя может выиграть, удвоив количество камней в куче, если $S = 18, \dots, 34$. При меньших значениях S за один ход нельзя получить кучу, в которой не менее 35 камней.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 17$ камней. Тогда после первого хода Пети в куче будет 18, 20 камней или 34 камня. Во всех случаях Ваня удваивает количество камней и выигрывает в один ход.

Задание 2

Возможные значения S : 14, 16. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 17 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (т.е. Петя) следующим ходом выигрывает.

Задание 3

Возможные значения S : 13, 15.

Например, для $S = 13$ после первого хода Пети в куче будет 14, 16 или 26 камней. Если в куче станет 26 камней, Ваня удвоит количество камней и выигрывает первым ходом. Ситуация, когда в куче 14 или 16 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани для первого возможного значения. Для второго возможного значения дерево строится аналогично. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов					
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)	
13			$17 + 1 = 18$	$18 * 2 = 36$	
			$17 + 3 = 20$	$20 * 2 = 40$	
	$13 + 1 = 14$	$14 + 3 = 17$	$17 * 2 = 34$	$34 + 1 = 35$	
			$17 * 2 = 34$	$34 + 3 = 37$	
				$34 * 2 = 68$	
			$17 + 1 = 18$	$18 * 2 = 36$	
			$17 + 3 = 20$	$20 * 2 = 40$	
	$13 + 3 = 16$	$16 + 1 = 17$		$34 + 1 = 35$	
			$17 * 2 = 34$	$34 + 3 = 37$	
				$34 * 2 = 68$	
		$13 * 2 = 26$	$26 * 2 = 52$		

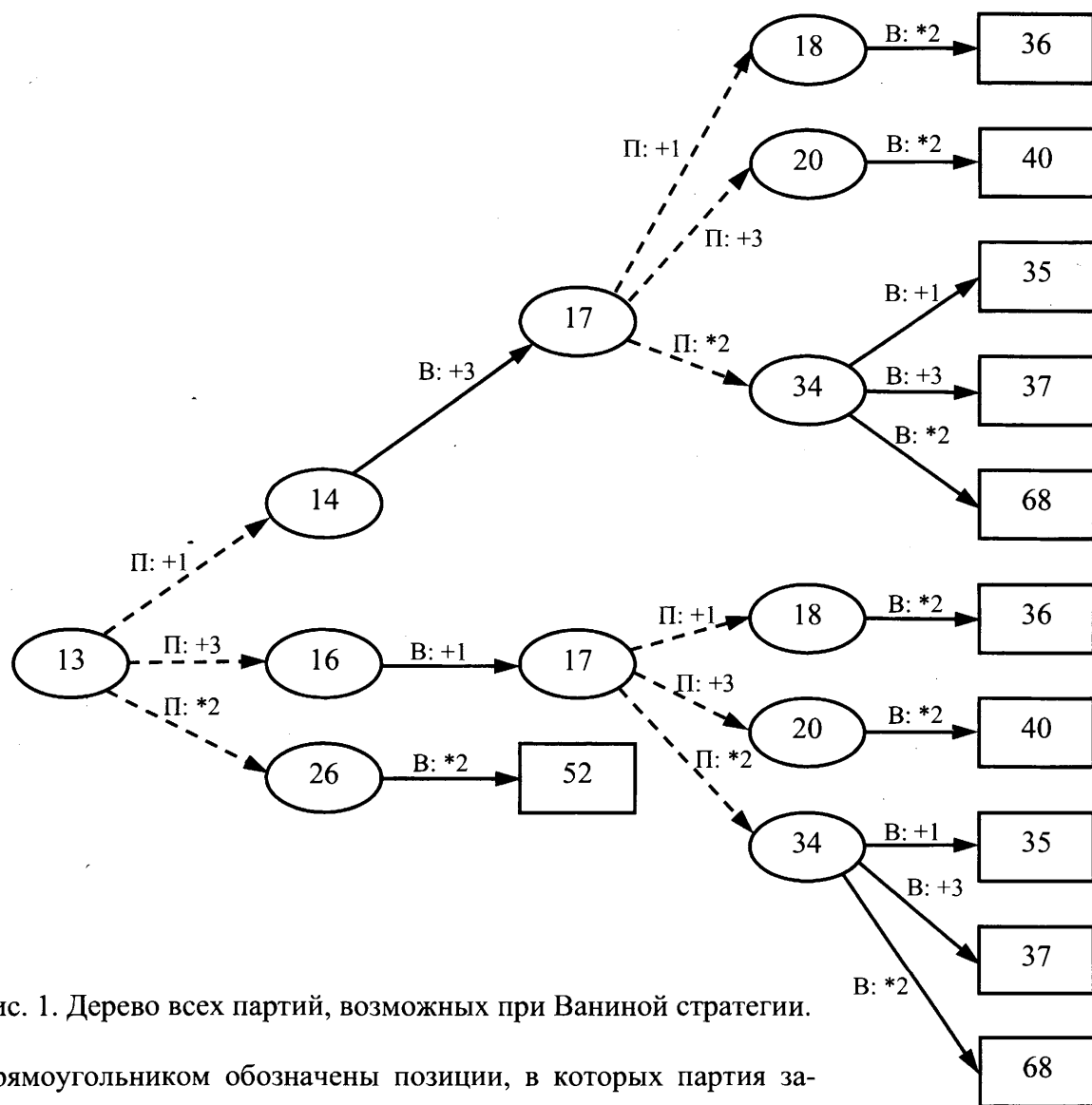


Рис. 1. Дерево всех партий, возможных при Ваниной стратегии.

Прямоугольником обозначены позиции, в которых партия заканчивается

При подготовке к ЕГЭ-2014 также желательно уделить внимание вариантам задач С3 прошлых лет.

Пример (ОС):

Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 2, а во второй – 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или утраивает число камней в какой-то куче, или добавляет 4 камня в какую-то кучу. Игра завершается в тот момент, когда общее число камней в двух кучах становится не менее 30. Если в момент завершения игры общее число камней в двух кучах не менее 41, то выиграл Петя, в противном случае – Ваня. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Решение:

Выигрывает Ваня.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделённые запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучах соответственно.

	1 ход	2 ход	3 ход	4 ход	
Стартовая позиция	Петя (все варианты хода)	Ваня (выигрышные ходы)	Петя (все варианты хода, кроме непосредственно проигрышных)	Ваня (выигрышные ходы, эквивалентному достаточно указать один из вариантов)	
2, 3	2, 9	<u>6, 9</u>	18, 9	<u>22, 9</u>	
				<u>18, 13</u>	
			10, 9	<u>30, 9</u>	
					<u>10, 27</u>
				6, 13	<u>18, 13</u>
	2, 7	<u>6, 7</u>		18, 7	<u>18, 21</u>
				6, 21	<u>18, 21</u>
					<u>6, 25</u>
				10, 7	<u>30, 7</u>
					<u>10, 21</u>
			6, 11	<u>6, 33</u>	
6, 3	<u>6, 9</u>	Те же варианты 3-4 хода			
					<u>6, 7</u>

Таблица содержит все возможные варианты ходов первого игрока. Из неё видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.

Задачи для самостоятельного решения (ОС)

26.1

Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 2, а во второй – 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или утраивает число камней в какой-то куче, или добавляет 4 камня в какую-то кучу. Игра завершается в тот момент, когда общее число камней в двух кучах становится не менее 30. Если в момент завершения игры количество камней в одной из куч не менее 34, то выиграл Ваня, в противном случае – Петя. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

26.2

Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 3, а во второй – 2 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или удваивает число камней в какой-то куче, или добавляет 3 камня в какую-то кучу. Игра за-

вершается в тот момент, когда количество камней в одной из куч становится не менее 15. Если в момент завершения игры количество камней в одной из куч не менее 19, то выиграл Ваня, в противном случае – Петя. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

26.3

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** камень или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 39. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 39 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 38$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрывающий ход для каждого указанного значения S .

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два таких значения S , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

26.4

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 44. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 44 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 43$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрывающий ход для каждого указанного значения S .

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два таких значения S , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

5.5

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 31. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 31 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 30$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрывающий ход для каждого указанного значения S .

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два таких значения S , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

Задание 14 проверяет умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд. Необходимо оценить результат исполнения алгоритма для различных наборов исходных данных. В некоторых случаях количество исходных наборов конечно и невелико, что позволяет решить задание «в лоб», простым перебором. В других случаях простым перебором задача не решается. Однако для любого задания этого типа существует непереборное решение, резко сокращающее количество исполнений алгоритма.

Пример (демоверсия 2010 г.):

Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
-----------------	----------------	----------------	-----------------

Цикл

ПОКА < условие > команда

выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнет движение в сторону стены, то он разрушится и программа прервется.

Сколько клеток лабиринта соответствуют требованию, что выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

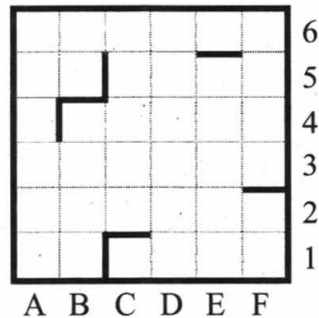
ПОКА < сверху свободно > вправо

ПОКА < справа свободно > вниз

ПОКА < снизу свободно > влево

ПОКА < слева свободно > вверх

КОНЕЦ



Решение:

Рассмотрим первый цикл: ПОКА < сверху свободно > вправо. Чтобы Робот не разрушился, необходимо чтобы он начинал путь из клетки с верхней стеной (тогда цикл не выполнится ни разу), либо, чтобы он дошел до такой клетки на своем пути вправо.

Этому требованию удовлетворяют клетки A2–F2, A6–F6, B4, C1. Если Робот начнет двигаться из одной из клеток A2–F2, то цикл закончится в клетке F2. Если из C5–E5, то цикл закончится в E5. Если из A6–F6, B4, или C1, то цикл не выполнится ни разу и Робот останется на месте. Итак, возможные конечные клетки первого цикла: F2, E5, A6–F6, B4, C1.

Рассмотрим второй цикл: ПОКА < справа свободно > вниз. Чтобы Робот не разрушился, необходимо чтобы он начинал выполнение этого цикла из клетки с правой стеной (тогда цикл не выполнится ни разу), либо, чтобы он дошел до такой клетки на своем пути вниз.

Из уже отобранных конечных клеток первого цикла, этому требованию удовлетворяют клетки A6, B6, F2, F6 и B4. При этом цикл закончится в клетках A4, B5, F2, F6 и B1 соответственно.

Итак, возможные маршруты первых двух циклов

- 1) A6 → A6 → A4
- 2) B6 → B6 → B5
- 3) F2 → F2 → F2
- 4) F6 → F6 → F6
- 5) B4 → B4 → B1

Выясним, куда приведет Робота третий цикл: ПОКА < снизу свободно > влево.

1) A6 → A6 → A4 → разрушение

2) B6 → B6 → B5 → B5

3) F2 → F2 → F2 → разрушение

4) F6 → F6 → F6 → E6

5) B4 → B4 → B1 → B1

Выясним, куда приведет Робота последний четвертый цикл:

ПОКА < слева свободно > вверх.

1) B6 → B6 → B5 → B5 → разрушение

2) F6 → F6 → F6 → E6 → разрушение

3) B4 → B4 → B1 → B1 → B4 – соответствует условию задачи.

Итак, мы нашли одну клетку, которая одновременно является начальной и конечной, показали, что других таких клеток нет.

Ответ: 1.

14 (демоверсия 2015 г.):

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду **сместиться на (a, b)** , где a, b – целые числа. Эта команда перемещает Чертёжника из точки с координатами (x, y) в точку с координатами

$(x + a; y + b)$.

Например, если Чертёжник находится в точке с координатами $(4, 2)$, то команда **сместиться на $(2, -3)$** переместит Чертёжника в точку $(6, -1)$.

Цикл

ПОВТОРИ *число* РАЗ

последовательность команд

КОНЕЦ ПОВТОРИ

означает, что *последовательность команд* будет выполнена указанное *число* раз (число должно быть натуральным).

Чертёжнику был дан для исполнения следующий алгоритм (буквами n, a, b обозначены неизвестные числа, при этом $n > 1$):

НАЧАЛО

сместиться на $(-3, -3)$

ПОВТОРИ n РАЗ

сместиться на (a, b)

сместиться на $(27, 12)$

КОНЕЦ ПОВТОРИ

сместиться на $(-22, -7)$

КОНЕЦ

Укажите наименьшее возможное значение числа n , для которого найдутся такие значения чисел a и b , что после выполнения программы Чертежник возвратится в исходную точку.

Решение:

По условию задачи Чертежник возвращается в ту же точку, из которой начал движение. Две команды исполняются им однократно: **сместиться на (-3, -3)** и **сместиться на (-22, -7)**, а две команды; **сместиться на (a, b)**, **сместиться на (27, 12)** – исполняются n раз. Возвращение в ту же точку означает итоговое смещение на (0,0). Тем самым должна выполняться система уравнений:

$$-3+n(a+27) - 22 = 0$$

$$-3+n(b+12) - 7 = 0$$

Выделив n , получаем:

$$n = 25 / (a + 27)$$

$$n = 10 / (b + 12)$$

Понятно, что если a и b – целые, то минимальное $n > 1$ будет равно 5 (наименьший общий делитель чисел 10 и 25). В этом случае $a = -22, b = -10$. Весь алгоритм примет вид:

НАЧАЛО

сместиться на (-3, -3)

ПОВТОРИ 5 РАЗ

сместиться на (-22, -10)

сместиться на (27, 12)

КОНЕЦ ПОВТОРИ

сместиться на (-22, -7)

КОНЕЦ

Проверим наше решение формальным исполнением этого алгоритма. Чертежник вернулся в ту же точку, в которой начал исполнение алгоритма.

Ответ: 5.

Задачи для самостоятельного решения (ОС)

14.1

Для заданий 14.1-14.5 система команд **РОБОТА** такая же, как и в разобранным выше примере.

Сколько клеток лабиринта соответствуют требованию, что, выполнив предложенную программу, **РОБОТ** уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

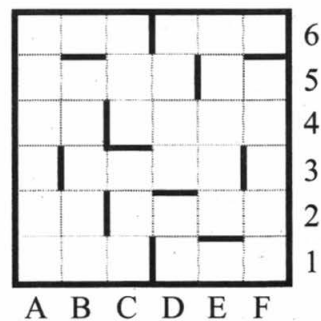
ПОКА < слева свободно > влево

ПОКА < снизу свободно > вниз

ПОКА < справа свободно > вправо

ПОКА < сверху свободно > вверх

КОНЕЦ



- 1) 5 2) 2 3) 3 4) 4

14.2

Сколько клеток лабиринта соответствуют требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

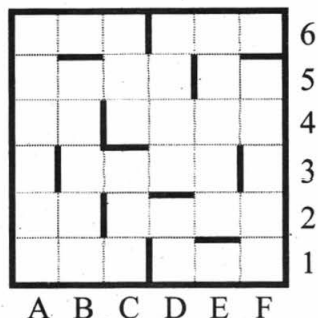
ПОКА < справа свободно > вниз

ПОКА < снизу свободно > влево

ПОКА < слева свободно > вверх

ПОКА < сверху свободно > вправо

КОНЕЦ



- 1) 1 2) 2 3) 3 4) 0

14.3

Сколько клеток лабиринта соответствуют требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

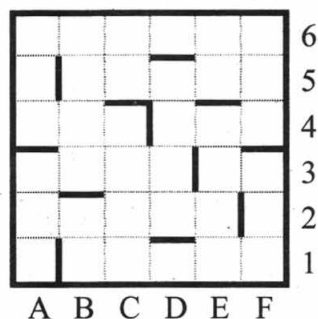
ПОКА < снизу свободно > вниз

ПОКА < слева свободно > влево

ПОКА < сверху свободно > вверх

ПОКА < справа свободно > вправо

КОНЕЦ



- 1) 1 2) 2 3) 3 4) 0

14.4

Сколько клеток лабиринта соответствуют требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

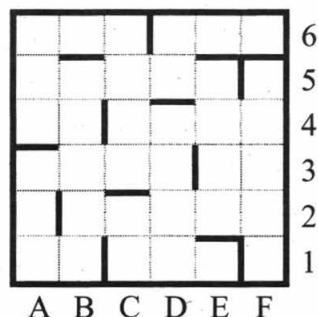
ПОКА < справа свободно > вправо

ПОКА < сверху свободно > вверх

ПОКА < слева свободно > влево

ПОКА < снизу свободно > вниз

КОНЕЦ



- 1) 5 2) 2 3) 3 4) 4

14.5

Сколько клеток лабиринта соответствуют требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

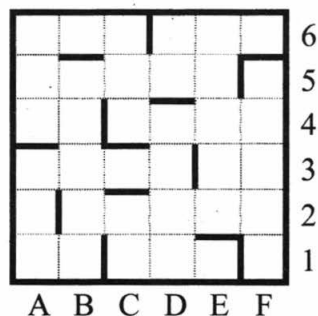
ПОКА < сверху свободно > вправо

ПОКА < справа свободно > вниз

ПОКА < снизу свободно > влево

ПОКА < слева свободно > вверх

КОНЕЦ



- 1) 1 2) 2 3) 5 4) 0

14.6

Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, включает в себя 4 команды-приказа и 4 команды проверки условия.

Команды-приказы:

вверх	вниз	влево	вправо
--------------	-------------	--------------	---------------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится, и программа прервётся.

Другие 4 команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
------------------------	-----------------------	-----------------------	------------------------

Цикл

ПОКА *условие*

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

ТО *команда1*

ИНАЧЕ *команда2*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно) или *команда2* (если условие ложно).

Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка А6)?

НАЧАЛО

ПОКА **снизу свободно** ИЛИ **слева свободно**

ЕСЛИ **снизу свободно**

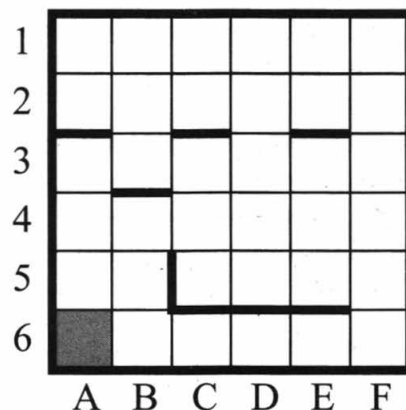
ТО **вниз**

ИНАЧЕ **влево**

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ



1) 15

2) 21

3) 23

4) 27

14.7

Для задания 14.7 система команд РОБОТА такая же, как и в задании 14.6.

Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка А6)?

НАЧАЛО

ПОКА слева свободно ИЛИ снизу свободно

ЕСЛИ слева свободно

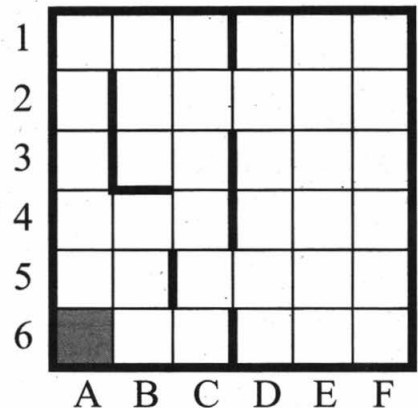
ТО влево

ИНАЧЕ вниз

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ



- 1) 14 2) 18 3) 23 4) 27

Задание 11 проверяет умение исполнить рекурсивный вычислительный алгоритм. Рекурсивным называется алгоритм, вызывающий в процессе исполнения сам себя. Для того, чтобы рекурсивный алгоритм имел завершение, требуется, чтобы его параметр изменялся в процессе исполнения и чтобы было явно написано условие завершения рекурсии. Задание имеет базовый уровень сложности и решается «в лоб», формальным исполнением рекурсивного алгоритма для данного значения параметра.

11 (В6 демоверсии 2013 г.)

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) * n, \text{ при } n > 1$$

Чему равно значение функции $F(5)$?

В ответе запишите только натуральное число.

Решение:

Запишем значение функции F для первых пяти натуральных чисел:

$$F(1) = 1$$

$$F(2) = 1 * 2 = 2$$

$$F(3) = 2 * 3 = 6$$

$$F(4) = 6 * 4 = 24$$

$$F(5) = 24 * 5 = 120$$

Нетрудно заметить, что $F(n) = 1 * 2 * \dots * n = n!$

Ответ: 120

Пример 11 (ОС 2014 г.)

Ниже на четырёх языках программирования записан рекурсивный алгоритм F.

Бейсик	Паскаль
<pre>SUB F(n) PRINT n IF n < 6 THEN F(n + 2) F(n * 3) END IF END SUB</pre>	<pre>procedure F(n: integer); begin writeln(n); if n < 6 then begin F(n + 2); F(n * 3) end end</pre>
Си	Алгоритмический язык
<pre>void F(int n) { printf("%d\n", n); if (n < 6) { F(n + 2); F(n * 3); } }</pre>	<pre>алг F(цел n) нач вывод n, нс если n < 6 то F(n + 2) F(n * 3) все кон</pre>

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(1)?

Решение:

Эта задача решается «в лоб», формальным исполнением рекурсивного алгоритма. Нетрудно заметить, что значение n печатается в начале выполнения рекурсивной функции. Таким образом, в начале будет напечатано число 1. Поскольку $1 < 6$, произойдет вызов функции F для аргумента $1+2=3$. Так что следующим напечатанным числом будет 3. Аналогично будет напечатано 5 и 7. При исполнении функции с аргументом 7 условие $7 < 6$ будет ложным, поэтому рекурсивные вызовы осуществляться не будут и функция вернется к исполнению с аргументом 5. Произойдет вызов функции F(15), которая напечатает число 15 и вернется обратно. Вызов функции F(5) завершится. Программа вернется к выполнению функции F(3) и вызовет функцию F(9): напечатает число 9 и вернется. Выполнение функции F(3) завершится. Программа вернется к выполнению функций F(1) и вызовет функцию F(3). Соответственно будет напечатано 3, 5, 7, 15, 9. После этого выполнение программы завершится. Таким образом, будет напечатано: 1, 3, 5, 7, 15, 9, 3, 5, 7, 15, 9. Сумма этих чисел: 79.

Ответ: 79

Задачи для самостоятельного решения (ОС)

11.1

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = 1 \text{ при } n \leq 2;$$

$$F(n) = F(n - 1) + 2 \times F(n - 2) \text{ при } n > 2.$$

Чему равно значение функции $F(7)$?

В ответе запишите только натуральное число.

11.2

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = 2 \text{ при } n \leq 2;$$

$$F(n) = F(n - 1) + 3 \times F(n - 2) \text{ при } n > 2.$$

Чему равно значение функции $F(5)$?

В ответе запишите только натуральное число.

11.3

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = n \text{ при } n \leq 2;$$

$$F(n) = F(n - 1) + 3 \times F(n - 2) \text{ при } n > 2.$$

Чему равно значение функции $F(6)$?

В ответе запишите только натуральное число.

11.4

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = n + 1 \text{ при } n \leq 2;$$

$$F(n) = F(n - 1) + 2 \times F(n - 2) \text{ при } n > 2.$$

Чему равно значение функции $F(4)$?

В ответе запишите только натуральное число.

11.5

Ниже на четырёх языках программирования записан рекурсивный алгоритм F.

Бейсик	Паскаль
<pre> SUB F(n) PRINT n IF n < 5 THEN F(n + 1) F(n * 2) END IF END SUB </pre>	<pre> procedure F(n: integer); begin writeln(n); if n < 5 then begin F(n + 1); F(n * 2) end end </pre>
Си	Алгоритмический язык
<pre> void F(int n) { printf("%d\n", n); if (n < 5) { F(n + 1); F(n * 2); } } </pre>	<pre> алг F(цел n) нач вывод n, нс если n < 5 то F(n + 1) F(n * 2) все кон </pre>

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова $F(2)$?

11.6

Ниже на четырёх языках программирования записан рекурсивный алгоритм F.

Бейсик	Паскаль
<pre>SUB F(n) PRINT n IF n > 0 THEN F(n - 3) F(n \ 3) END IF END SUB</pre>	<pre>procedure F(n: integer); begin writeln(n); if n > 0 then begin F(n - 3); F(n div 3) end end</pre>
Си	Алгоритмический язык
<pre>void F(int n) { printf("%d\n", n); if (n > 0) { F(n - 3); F(n / 3); } }</pre>	<pre><u>алг</u> F(<u>цел</u> n) <u>нач</u> <u>ВЫВОД</u> n, <u>НС</u> <u>если</u> n > 0 <u>то</u> F(n - 3) F(div(n, 3)) <u>все</u> <u>кон</u></pre>

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(9)?

2.2. Программирование

Для решения задач по этой теме требуется знание языка программирования. Для выполнения всех заданий, кроме С4, достаточно знания основных управляющих конструкций и типов данных хотя бы одного из языков – Си, Паскаля, Бейсика или Алгоритмического языка.

Простейшим заданием на эту тему в 2014 г. было задание В2. Оно проверяло знание оператора ветвления и свойств арифметических операций с данными целого типа (приоритет операций, операции деления нацело и получения остатка). В демоверсии 2015 г. такого задания нет, но для тренировки сделать его следует.

B2 (OC)

Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre>a = 40 b = 10 b = - a / 2 * b IF a < b THEN c = b - a ELSE c = a - 2 * b ENDIF</pre>	<pre>a := 40; b := 10; b := - a / 2 * b; if a < b then c := b - a else c := a - 2 * b;</pre>
Си	Алгоритмический язык
<pre>a = 40; b = 10; b = - a / 2 * b; if (a < b) c = b - a; else c = a - 2 * b;</pre>	<pre>a := 40 b := 10 b := - a / 2 * b если a < b то c := b - a иначе c := a - 2 * b все</pre>

Решение:

Порядок выполнения операций при вычислении целочисленных выражениях языка программирования такой же, как и в обычной арифметике:

- сначала вычисляется значение формулы в скобках (если они есть);
- затем выполняются операции умножения, деления, взятия остатка;
- в последнюю очередь выполняются операции сложения и вычитания;
- унарный минус (-...) сокращением записи $-1 * \dots$.

После присваивания $b = - a / 2 * b$, значение b станет равным

$$-40 / 2 * 10 = -200.$$

Значение выражения $a < b$ – ложь, поэтому в условном операторе выполнится ветвь ELSE (иначе) т.е. $c = a - 2 * b$, значение c станет равным $40 - 2 * (-200) = 440$.

Ответ: 440

Задачи для самостоятельного решения (ОС)

В2.1

Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre>a = 20 b = 5 b = - a / 2 * b IF a < b THEN c = a + b ELSE c = a + 2 * b ENDIF</pre>	<pre>a := 20; b := 5; b := - a / 2 * b; if a < b then c := a + b else c := a + 2 * b;</pre>
Си	Алгоритмический
<pre>a = 20; b = 5; b = - a / 2 * b; if (a < b) c = a + b; else c = a + 2 * b;</pre>	<pre>a := 20 b := 5 b := - a / 2 * b если a < b то c := a + b иначе c := a + 2 * b все</pre>

В2.2

Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre>a = 30 b = 5 b = - a / 3 * b IF a > b THEN c = a + b ELSE c = a + 2 * b ENDIF</pre>	<pre>a := 30; b := 5; b := - a / 3 * b; if a > b then c := a + b else c := a + 2 * b;</pre>
Си	Алгоритмический
<pre>a = 30; b = 5; b = - a / 3 * b; if (a > b) c = a + b; else c = a + 2 * b;</pre>	<pre>a := 30 b := 5 b := - a / 3 * b если a > b то c := a + b иначе c := a + 2 * b все</pre>

В2.3

Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre>a = 30 b = 3 b = - a / 2 * b IF a < b THEN c = a + b ELSE c = a + 2 * b ENDIF</pre>	<pre>a := 30; b := 3; b := - a / 2 * b; if a < b then c := a + b else c := a + 2 * b;</pre>
Си	Алгоритмический
<pre>a = 30; b = 3; b = - a / 2 * b; if (a < b) c = a + b; else c = a + 2 * b;</pre>	<pre>a := 30 b := 3 b := - a / 2 * b если a < b то c := a + b иначе c := a + 2 * b все</pre>

В2.4

Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

Бейсик	Паскаль
<pre>a = 40 b = 5 a = a / 2 * b IF a < b THEN c = a - 2 * b ELSE c = a + 2 * b ENDIF</pre>	<pre>a := 40; b := 5; a := a / 2 * b; if a < b then c := a - 2 * b else c := a + 2 * b;</pre>
Си	Алгоритмический язык
<pre>a = 40; b = 5; a = a / 2 * b; if (a < b) c = a - 2 * b; else c = a + 2 * b;</pre>	<pre>a := 40 b := 5 a := a / 2 * b если a < b то c := a - 2 * b иначе c := a + 2 * b все</pre>

Задание 8 демоверсии 2015 г. соответствует заданию В5 в ЕГЭ 2014 г. Оно немного сложнее задания В2, так как содержит конструкцию цикла. При подготовке к экзамену следует учитывать, что в реальном экзамене 2015 г. на этой позиции могут быть задания, проверяющие как знание конструкции цикла, так и конструкции ветвления.

Пример задания 8 (из демоверсии 2012 г., тогда задание стояло на позиции В3)

Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 0 k = 0 WHILE s < 1024 s = s + 10 k = k + 1 WEND PRINT k</pre>	<pre>Var k, s : integer; BEGIN s:=0; k:=0; while s<1024 do begin s:=s+10; k:=k+1; end; write(k); END.</pre>
Си	Алгоритмический
<pre>{ int k, s; s = 0; k = 0; while (s<1024) { s = s+10; k = k+1; } printf("%d", k); }</pre>	<pre><u>нач</u> <u>цел</u> k, s. s:=0 k:=0 <u>нц пока</u> s < 1024 s:=s+10; k:=k+1 <u>кц</u> <u>вывод</u> k <u>кон</u></pre>

Решение:

При решении этой задачи конечно же не требуется непосредственное выполнение алгоритма. Обозначим S_k значение переменной s после выполнения k -й итерации цикла. Значения индекса k будет совпадать со значением переменной k в программе. Условие прекращения цикла тогда можно будет записать в виде $S_k \geq 1024$. Найдем минимальное S_k , удовлетворяющее этому условию. Из соотношений $S_0=0$, $S_k=S_{k-1}+10$ ($k=1, 2, \dots$) следует, что $S_k = k \cdot 10$. Поэтому условие прекращения цикла выполнится при $S_k = 1030$. Таким образом, на момент выхода из цикла $k = 1030/10 = 103$.

Ответ: 103.

Задачи для самостоятельного решения

8.1

Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 1024 k = 1 WHILE s >= 1 s = s \ 2 k = k * 2 WEND PRINT k</pre>	<pre>Var k, s : integer; BEGIN s:=1024; k:=1; while s >= 1 do begin s:=s div 2; k:=k * 2; end; write(k); END.</pre>
Си	Алгоритмический
<pre>{ int k, s; s = 1024; k = 1; while (s>=1) { s = s/2; k = k*2; } printf("%d", k); }</pre>	<pre><u>нач</u> <u>цел</u> k, s s:=1024 k:=1 <u>нц пока</u> s >= 1 s:= div (s, 2); k:= k*2 <u>кц</u> <u>ВЫВОД</u> k <u>кон</u></pre>

8.2

Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 8 k = 1 WHILE s < 2012 s = s * 2 k = k + 1 WEND PRINT k</pre>	<pre>Var k, s : integer; BEGIN s:=8; k:=1; while s < 2012 do begin s:=s * 2; k:=k+1; end; write(k); END.</pre>
Си	Алгоритмический
<pre>{ int k, s; s = 8; k = 1; while (s<2012) { s = s*2; k = k+1; } printf("%d", k); }</pre>	<pre><u>нач</u> <u>цел</u> k, s s:=8 k:=1 <u>нц пока</u> s < 2012 s:= s*2; k:=k+1 <u>кц</u> <u>ВЫВОД</u> k <u>кон</u></pre>

8.3

Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 1024 k = 64 WHILE s >=1 s = s \ 2 k = k - 1 WEND PRINT k</pre>	<pre>Var k, s : integer; BEGIN s:=1024; k:=64; while s>=1 do begin s:=s div 2; k:=k-1; end; write(k); END.</pre>
Си	Алгоритмический
<pre>{ int k, s; s = 1024; k = 64; while (s>=1) { s = s/2; k = k-1; } printf("%d", k); }</pre>	<pre>нач цел k, s s:=1024 k:=64 нц пока s >= 1 s:= div(s,2); k:=k-1 кц вывод k кон</pre>

8.4

Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM a, b AS INTEGER a = 91 b = 65 WHILE a <> b IF a > b THEN a = a - b ELSE b = b - a WEND PRINT a</pre>	<pre>Var a, b : integer; BEGIN a:=91; b:= 65; while a <> b do if a > b then a := a - b else b := b - a; write(a); END.</pre>
Си	Алгоритмический
<pre>{ int a, b; a:=91; b:= 65; while (a!=b) { if (a > b) a = a - b; else b = b - a; } printf("%d", a); }</pre>	<pre>нач цел a, b a:=91 b:= 65 нц пока a <> b если a > b то a := a - b иначе b := b - a все кц вывод a кон</pre>

8.5

Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM p, q AS INTEGER p = 900 q = 21 WHILE p <> q IF p < q THEN q = q - p ELSE p = p - q WEND PRINT q</pre>	<pre>Var p, q : integer; BEGIN p:=900; q:= 21; while p <> q do if p < q then q := q - p else p := p - q; write(q); END.</pre>
Си	Алгоритмический
<pre>{ int p, q; p:=900; q:= 21; while (p!=q) { if (p < q) q = q - p; else p = p - q; } printf("%d", q); }</pre>	<pre>нач цел p, q p:=900 q:= 21 нц пока p <> q если p < q то q := q - p иначе p := p - q все кц Вывод q кон</pre>

8.6

Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM p, q AS INTEGER p = 2100 q = 3200 WHILE p > 200 p = p-1 q = q+1 WEND PRINT q</pre>	<pre>Var p, q : integer; BEGIN p:=2100; q:= 3200; while p > 200 do begin p := p - 1; q := q + 1; end; write(q); END.</pre>
Си	Алгоритмический
<pre>{ int p, q; p:=2100; q:=3200; while (p>200) { p = p - 1; q = q + 1; } printf("%d", q); }</pre>	<pre>нач цел p, q p:= 2100 q:= 3200 нц пока p > 200 p:=p-1 q:=q+1 кц вывод q кон</pre>

Задание 19 демоверсии 2015 г. соответствует заданию A12 в ЕГЭ 2014 г. Оно проверяет умение работать с массивами. Следует обратить внимание, что согласно спецификации может проверяться знание любых алгоритмов работы с массивами, в том числе сортировки. На практике задания ЕГЭ обычно содержат алгоритмы заполнения массива, а затем перестановки значений в массиве.

19 (A12 демоверсии 2010 г.)

В программе используется одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 10 A(i)=i NEXT i FOR i=0 TO 10 A(10-i)=A(i) A(i)=A(10-i) NEXT i</pre>	<pre>for i:=0 to 10 do A[i]:=i; for i:=0 to 10 do begin A[10-i]:=A[i]; A[i]:=A[10-i]; end;</pre>
Си	Алгоритмический язык
<pre>for (i=0;i<=10;i++) A[i]=i; for (i=0;i<=10;i++) { A[10-i]=A[i]; A[i]=A[10-i]; }</pre>	<pre><u>нц</u> <u>для</u> i <u>от</u> 0 <u>до</u> 10 A[i]:=i <u>кц</u> <u>нц</u> <u>для</u> i <u>от</u> 0 <u>до</u> 10 A[10-i]:=A[i] A[i]:=A[10-i] <u>кц</u></pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 10 9 8 7 6 5 4 3 2 1 0
- 2) 0 1 2 3 4 5 6 7 8 9 10
- 3) 10 9 8 7 6 5 6 7 8 9 10
- 4) 0 1 2 3 4 5 4 3 2 1 0

Решение:

Найдем значения элементов массива после выполнения первого цикла

```
FOR i=0 TO 10
A(i)=i
NEXT i
```

0 1 2 ... i ... 8 9 10

<i>0</i>	<i>1</i>	<i>2</i>	...	<i>i</i>	...	<i>8</i>	<i>9</i>	<i>10</i>
----------	----------	----------	-----	----------	-----	----------	----------	-----------

Сверху приводятся индексы элементов массива, ниже – значения элементов. Значения элементов выделены курсивом.

Теперь найдем значения элементов массива после выполнения второго цикла.

```
FOR i=0 TO 10
A(10-i)=A(i)
A(i)=A(10-i)
NEXT i
```

Заметим, что оператор $A(i)=A(10-i)$ является лишним, т.к. предыдущий оператор уже обеспечивает равенство i -го и $10-i$ -го элементов.

Поэтому цикл можно переписать в виде

```
FOR i=0 TO 10
A(10-i)=A(i)
NEXT i
```

После выполнения цикла массив примет вид:

0	1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	4	3	2	1	0

В общем виде:

0	1	2	...	i	...	8	9	10
0	1	2	...	10 - i при $i \geq 5$, i при $i < 5$...	2	1	0

Правильен 4-й вариант ответа.

Ответ: 4.

19 (A12 демоверсия 2012 г.)

В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 9 A(i)=9-i NEXT i FOR i=0 TO 4 k = A(i) A(i) = A(9-i) A(9-i) =k NEXT i</pre>	<pre>for i:=0 to 9 do A[i]:=9-i; for i:=0 to 4 do begin k:=A[i]; A[i]:=A[9-i]; A[9-i]:=k; end;</pre>
Си	Алгоритмический язык
<pre>for (i=0; i<=9; i++) A[i]=9-i; for (i=0; i<=4; i++) { k=A[i]; A[i]=A[9-i]; A[9-i]=k; }</pre>	<pre>нц для i от 0 до 9 A[i]:=9-i кц нц для i от 0 до 4 k:=A[i] A[i]:=A[9-i] A[9-i]:=k кц</pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 9 8 7 6 5 4 3 2 1 0
- 2) 0 1 2 3 4 5 6 7 8 9
- 3) 9 8 7 6 5 5 6 7 8 9
- 4) 0 1 2 3 4 4 3 2 1 0

Решение:

В цикле

нц для i от 0 до 9

$A[i] := 9 - i$

кц

элементы массива инициализируются следующим образом:

$A[0] = 9, A[1] = 8, \dots, A[i] = 9 - i, \dots, A[9] = 0.$

Тело второго цикла

$k := A[i]$

$A[i] := A[9 - i]$

$A[9 - i] := k$

реализует обмен значениями элементов $A[i]$ и $A[9 - i]$, т.е. элементов расположенных симметрично относительно середины массива. Поскольку этот цикл выполняется для i от 0 до 4, то массив будет изменен следующим образом:

до начала второго цикла:

9	8	7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---	---

1-я итерация (для $i = 0$):

0	8	7	6	5	4	3	2	1	9
---	---	---	---	---	---	---	---	---	---

2-я итерация (для $i = 1$):

0	1	7	6	5	4	3	2	8	9
---	---	---	---	---	---	---	---	---	---

5-я и последняя итерация (для $i = 4$)

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

Ответ: 2.

Задачи для самостоятельного решения (ОС)

19.1

В программе используется одномерный целочисленный массив А с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 9 A(i)=i NEXT i FOR i=9 TO 0 STEP -1 A(9-i)=A(i) NEXT i</pre>	<pre>for i:=0 to 9 do A[i]:=i; for i:=9 downto 0 do A[9-i]:=A[i];</pre>
Си	Алгоритмический язык
<pre>for (i=0;i<=9;i++) A[i]=i; for (i=9;i>=0;i--) A[9-i]=A[i];</pre>	<pre>нц для i от 0 до 9 A[i]:=i кц нц для i от 9 до 0 шаг -1 A[9-i]:=A[i] кц</pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 9 8 7 6 5 4 3 2 1 0
- 2) 0 1 2 3 4 5 6 7 8 9
- 3) 9 8 7 6 5 5 6 7 8 9
- 4) 0 1 2 3 4 4 3 2 1 0

19.2

В программе используется одномерный целочисленный массив А с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 9 A(i)=i NEXT i FOR i=0 TO 9 A(i)=A(9-i) NEXT i</pre>	<pre>for i:=0 to 9 do A[i]:=i; for i:=0 to 9 do A[i]:=A[9-i];</pre>
Си	Алгоритмический язык
<pre>for (i=0;i<=9;i++) A[i]=i; for (i=0;i<=9;i++) A[i]=A[9-i];</pre>	<pre>нц для i от 0 до 9 A[i]:=i кц нц для i от 0 до 9 A[i]:=A[9-i] кц</pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 9 8 7 6 5 4 3 2 1 0
- 2) 0 1 2 3 4 5 6 7 8 9
- 3) 9 8 7 6 5 5 6 7 8 9
- 4) 0 1 2 3 4 4 3 2 1 0

19.3

В программе используется одномерный целочисленный массив А с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 9 A(i)=9-i NEXT i FOR i=0 TO 9 A(i)=A(9-i) NEXT i</pre>	<pre>for i:=0 to 9 do A[i]:=9-i; for i:=0 to 9 do A[i]:=A[9-i];</pre>
Си	Алгоритмический язык
<pre>for (i=0; i<=9; i++) A[i]=9-i; for (i=0; i<=9; i++) A[i]=A[9-i];</pre>	<pre><u>нц</u> <u>для</u> <u>i</u> <u>от</u> 0 <u>до</u> 9 A[i]:=9-i <u>кц</u> <u>нц</u> <u>для</u> <u>i</u> <u>от</u> 0 <u>до</u> 9 A[i]:=A[9-i] <u>кц</u></pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 9 8 7 6 5 4 3 2 1 0
- 2) 0 1 2 3 4 5 6 7 8 9
- 3) 9 8 7 6 5 5 6 7 8 9
- 4) 0 1 2 3 4 4 3 2 1 0

19.4

В программе используется одномерный целочисленный массив А с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 9 A(i)=9-i NEXT i FOR i=0 TO 4 A(i) = A(9-i) NEXT i</pre>	<pre>for i:=0 to 9 do A[i]:=9-i; for i:=0 to 4 do begin A[i]:=A[9-i]; end;</pre>
Си	Алгоритмический язык
<pre>for (i=0; i<=9; i++) A[i]=9-i; for (i=0; i<=4; i++) { A[i]=A[9-i]; }</pre>	<pre><u>нц</u> <u>для</u> i <u>от</u> 0 <u>до</u> 9 A[i]:=9-i <u>кц</u> <u>нц</u> <u>для</u> i <u>от</u> 0 <u>до</u> 4 A[i]:=A[9-i] <u>кц</u></pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 9 8 7 6 5 4 3 2 1 0
- 2) 4 3 2 1 0 0 1 2 3 4
- 3) 9 8 7 6 5 5 6 7 8 9
- 4) 0 1 2 3 4 4 3 2 1 0

19.5

В программе используется одномерный целочисленный массив А с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 9 A(i)=9-i NEXT i FOR i=0 TO 4 A(i) = A(i+1) NEXT i</pre>	<pre>for i:=0 to 9 do A[i]:=9-i; for i:=0 to 4 do begin A[i]:=A[i+1]; end;</pre>
Си	Алгоритмический язык
<pre>for (i=0; i<=9; i++) A[i]=9-i; for (i=0; i<=4; i++) { A[i]=A[i+1]; }</pre>	<pre><u>нц</u> <u>для</u> i <u>от</u> 0 <u>до</u> 9 A[i]:=9-i <u>кц</u> <u>нц</u> <u>для</u> i <u>от</u> 0 <u>до</u> 4 A[i]:=A[i+1] <u>кц</u></pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

1) 9 8 7 6 5 4 3 2 1 0

2) 8 7 6 5 4 3 2 1 0 0

3) 9 8 7 6 5 5 6 7 8 9

4) 8 7 6 5 4 4 3 2 1 0

Задание 20 (в ЕГЭ 2014 г. – В8)

Это задание относительно новое (до 2012 г. задания такого типа в ЕГЭ отсутствовали), поэтому при подготовке к экзамену ему следует уделить повышенное внимание. Особенностью данного задания является то, что в нем не задаются конкретные входные данные (значение x), поэтому найти ответ непосредственным исполнением программы невозможно. Для решения задач приведенного в демоверсии типа необходимо понимать смысл арифметических операций над целыми числами mod (в Си обозначается $\%$) и div (в Си обозначается $/$, в Бейсике \backslash).

Операция $A \text{ mod } B$ означает получение остатка от деления целого числа A на целое число B . Результат операции является целым числом, меньшим B . При $B=0$ результат операции не определен. Если B является делителем A , то остаток от деления A на B равен нулю, в частности $A \text{ mod } A = 0$. Если $A < B$, то $A \text{ mod } B = A$.

Примеры:

$$7 \text{ mod } 3 = 7 \text{ mod } 2 = 7 \text{ mod } 6 = 1$$

$$7 \text{ mod } 5 = 2$$

$$7 \text{ mod } 1 = 7 \text{ mod } 7 = 0$$

$$7 \text{ mod } 8 = 7 \text{ mod } 10000 = 7$$

$$12 \text{ mod } 12 = 12 \text{ mod } 6 = 12 \text{ mod } 4 = 12 \text{ mod } 3 = 12 \text{ mod } 2 = 12 \text{ mod } 1 = 0$$

$$13 \text{ mod } 12 = 13 \text{ mod } 6 = 13 \text{ mod } 4 = 13 \text{ mod } 3 = 13 \text{ mod } 2 = 1$$

Операция $A \text{ div } B$ (целочисленное деление) означает получение целого частного при делении A на B . При $B=0$ результат операции не определен. Целочисленное деление можно понимать как обычное деление вещественных чисел с отбрасыванием (не округлением!) дробной части результата. Если $A < B$, то результат операции всегда будет равен нулю.

Примеры:

$$7 \text{ div } 7 = 7 \text{ div } 6 = 7 \text{ div } 5 = 7 \text{ div } 4 = 1$$

$$7 \text{ div } 3 = 2$$

$$7 \text{ div } 2 = 3$$

$$7 \text{ div } 8 = 7 \text{ div } 10000 = 0$$

При вычислении арифметических выражений, операции div и mod имеют такой же приоритет, как и операции сложения и умножения.

Примеры:

$$3 + 5 \text{ mod } 2 = 4$$

$$(3 + 5) \text{ mod } 2 = 0$$

$$7 \text{ div } 3 \cdot 2 = 2 \cdot 7 \text{ div } 3 = 4$$

$$(7 \text{ div } 2) \cdot (7 \text{ div } 2) = 9$$

$$(7 \cdot 7) \text{ div } (2 \cdot 2) = 11$$

Для операций div и mod выполняется соотношение $A = (A \text{ div } B) \cdot B + A \text{ mod } B$.

Операции div и mod можно использовать для вычисления цифр числа в различных системах счисления. Остаток от деления A на B будет значением последней цифры числа A , представленного в системе счисления с основанием B .

Примеры:

$137 \text{ mod } 10 = 7$ (последняя цифра числа 137, записанного в системе счисления с основанием 10)

$9 \text{ mod } 2 = 1$ (последняя цифра числа 9, записанного в системе счисления с основанием 2; $9_{10} = 1001_2$)

$29 \text{ mod } 16 = 13$ (числовое значение последней цифры числа 29, записанного в системе счисления с основанием 16; $29_{10} = 1D_{16}$)

Результатом целочисленного деления $A \text{ div } B$ будет отбрасывание последней цифры числа A записанного в системе счисления с основанием B .

Примеры:

$137 \text{ div } 10 = 13$ (отброшена последняя цифра в записи числа 137 в системе счисления с основанием 10)

$9 \text{ div } 2 = 4 = 100_2$ (отброшена последняя цифра числа 9, записанного в системе счисления с основанием 2; $9_{10} = 1001_2$)

$29 \text{ div } 16 = 1$ (отброшена последняя цифра в записи числа 29 в системе счисления с основанием 16; $29_{10} = 1D_{16}$)

20 (из демоверсии 2012 г.)

Ниже на 4-х языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа L и M . Укажите наибольшее из таких чисел x , при вводе которых алгоритм печатает сначала 3, а потом 7.

Бейсик	Паскаль
<pre> DIM X, L, M AS INTEGER INPUT (X) L=0: M=0 WHILE X > 0 L = L+1 IF M < (X MOD 10) M = X MOD 10 ENDIF X = X \ 10 WEND PRINT L PRINT M </pre>	<pre> var x, L, M: integer; begin readln(x); L:=0; M:=0; while x>0 do begin L:=L+1; if M < (x mod 10) then begin M:=x mod 10; end end; x:= x div 10; end; writeln(L); write(M); end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int x, L, M; scanf("% d", &x); L=0; M=0; while (x>0){ L=L+1; if (M < x%10) { M = x%10 } x= x/10; } printf("%d\n%d", L, M); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> x, L, M <u>ввод</u> x L:=0; M:=0 <u>нц пока</u> x>0 L:=L+1 <u>если</u> M < mod(x,10) <u>то</u> M:= mod(x,10) <u>все</u> x:=div(x,10) <u>кц</u> <u>вывод</u> L, <u>нс</u>, M <u>кон</u> </pre>

Решение:

Выясним, что делает цикл

```
WHILE X > 0
  L = L+1
  IF M < (X MOD 10)
    M = X MOD 10
  ENDIF
  X = X \ 10
WEND
```

Каждое выполнение цикла «откусывает» справа от исходного числа по одной цифре в его десятичной записи ($X = X \setminus 10$), увеличивая на 1 счетчик «откушенных» цифр ($L = L+1$). Поскольку условие продолжения цикла $X > 0$, то цикл завершится, когда исходное число будет «съедено» целиком. При этом в переменной L сохранится количество «откушенных» цифр, т.е. сколько всего десятичных цифр было в записи числа.

По условию L после окончания цикла равно 3. Это возможно в том, и только том случае, когда x состоит ровно из трех значащих цифр. Итак, x – трехзначное число.

Теперь вернемся к значению M . Условие

```
IF M < (X MOD 10)
  M = X MOD 10
ENDIF
```

записанное внутри цикла, не что иное, как поиск максимума среди значений $X \text{ MOD } 10$, т.е. цифр числа x . По условию M равно 7, поэтому максимальной цифрой числа x является 7. Итак, число x трехзначно, любая его цифра не больше 7. Очевидно, максимальным числом с такими свойствами будет 777.

Ответ 777.

Пример:

Пусть в текстах программ в предыдущей задаче операторы взятия остатка и целочисленного деления $X \text{ MOD } 10$ и $X \setminus 10$ (и их аналоги на Паскале, Си и Алгоритмическом языке) заменены на $X \text{ MOD } 8$ и $X \setminus 8$ соответственно.

Укажите наибольшее из таких чисел x , при вводе которых алгоритм печатает сначала 3, а потом 7.

Решение:

Для этого случая справедливы практически все приведенные выше рассуждения. Отличие состоит в том, что цифры числа выделяются в восьмеричной системе счисления. Поэтому результат 777 нужно перевести из восьмеричной в десятичную систему счисления.

$$777_8 = 511_{10}$$

Ответ: 511.

Задачи для самостоятельного решения

20.1

Ниже на 4-х языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа L и M . Укажите наибольшее из таких чисел x , при вводе которых алгоритм печатает сначала 2, а потом 6.

Бейсик	Паскаль
<pre> DIM X, L, M AS INTEGER INPUT (X) L=0: M=0 WHILE X > 0 L = L+1 IF M < (X MOD 10) M = X MOD 10 ENDIF X = X \ 10 WEND PRINT L PRINT M </pre>	<pre> var x, L, M: integer; begin readln(x); L:=0; M:=0; while x>0 do begin L:=L+1; if M < (x mod 10) then begin M:=x mod 10; end x:= x div 10; end; writeln(L); write(M); end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int x, L, M; scanf("%d", &x); L=0; M=0; while (x>0){ L=L+1; if (M < x%10) { M = x%10 } x= x/10; } printf("%d\n%d", L, M); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> x, L, M <u>ввод</u> x L:=0; M:=0 <u>нц пока</u> x>0 L:=L+1 <u>если</u> M < mod(x,10) <u>то</u> M:= mod(x,10) <u>все</u> x:=div(x,10) <u>кц</u> <u>вывод</u> L, <u>нс</u>, M <u>кон</u> </pre>

20.2

Для алгоритма из предыдущей задачи укажите наименьшее из таких чисел x , при вводе которых печатается сначала 2, а потом 6.

20.3

Ниже на 4-х языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа L и M . Укажите наибольшее из таких чисел x , при вводе которых алгоритм печатает сначала 3, а потом 6.

Бейсик	Паскаль
<pre>DIM X, L, M AS INTEGER INPUT (X) L=0: M=9 WHILE X > 0 L = L+1 IF M > (X MOD 10) M = X MOD 10 ENDIF X = X \ 10 WEND PRINT L PRINT M</pre>	<pre>var x, L, M: integer; begin readln(x); L:=0; M:=9; while x>0 do begin L:=L+1; if M > (x mod 10) then begin M:=x mod 10; end end; x:= x div 10; end; writeln(L); write(M); end.</pre>

Си	Алгоритмический
<pre>#include<stdio.h> void main() { int x, L, M; scanf("% d", &x); L=0; M=9; while (x>0){ L=L+1; if (M > x%10) { M = x%10 } x= x/10; } printf("%d\n%d", L, M); }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> x, L, M <u>ввод</u> x L:=0; M:=9 <u>нц пока</u> x>0 L:=L+1 <u>если</u> M > mod(x,10) <u>то</u> M:= mod(x,10) <u>все</u> x:=div(x,10) <u>кц</u> <u>вывод</u> L, <u>нс</u>, M <u>кон</u></pre>

20.4

Для алгоритма из предыдущей задачи укажите наименьшее из таких чисел x , при вводе которых печатается сначала 3, а потом 5.

20.5

Для алгоритма из предыдущей задачи укажите наименьшее из таких чисел x , при вводе которых печатается сначала 4, а потом 0.

20.6

Для алгоритма из предыдущей задачи укажите наибольшее из таких чисел x , при вводе которых печатается сначала 4, а потом 0.

20.7

Ниже на 4-х языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа L и M . Укажите наибольшее из таких чисел x , при вводе которых алгоритм печатает сначала 3, а потом 4.

Бейсик	Паскаль
<pre> DIM X, L, M AS INTEGER INPUT (X) L=0: M=0 WHILE X > 0 L = L+1 IF M < (X MOD 5) M = X MOD 5 ENDIF X = X \ 5 WEND PRINT L PRINT M </pre>	<pre> var x, L, M: integer; begin readln(x); L:=0; M:=0; while x>0 do begin L:=L+1; if M < (x mod 5) then begin M:=x mod 5; end x:= x div 5; end; writeln(L); write(M); end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int x, L, M; scanf("% d", &x); L=0; M=0; while (x>0){ L=L+1; if (M < x%5) { M = x%5 } x= x/5; } printf("%d\n%d", L, M); } </pre>	<pre> <u>алг</u> <u>нач</u> цел x, L, M <u>ввод</u> x L:=0; M:=0 <u>нц пока</u> x>0 L:=L+1 <u>если</u> M < mod(x,5) <u>то</u> M:= mod(x,5) <u>все</u> x:=div(x,5) <u>кц</u> <u>вывод</u> L, <u>нс</u>, M <u>кон</u> </pre>

Задание 21 (в ЕГЭ 2014 г. – В14)

Этот тип заданий также является относительно новым. Оно проверяет умение анализировать программу, использующую процедуры и функции. Функция может многократно вызываться из основного алгоритма, ей могут передаваться данные из основного алгоритма, и она, в свою очередь, может возвращать значения в основной алгоритм. Конкретные соглашения о формате вызова вспомогательного алгоритма зависят от языка программирования. В задании демоверсии вспомогательным алгоритмом является функция F.

Бейсик	Паскаль
<pre>FUNCTION F (x) F = 4*(x - 1) * (x - 3) END FUNCTION</pre>	<pre>Function F(x:integer):integer; begin F:=4*(x-1)*(x-3); end;</pre>
Си	Алгоритмический
<pre>int F(int x) { return 4*(x-1)*(x-3); }</pre>	<pre><u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> := 4*(x-1)*(x-3) <u>кон</u></pre>

Как известно, в языках программирования используются стандартные функции, например, Sin для вычисления синуса. Вызов стандартной функции Sin выглядит (с точностью до синтаксических деталей языка) следующим образом: $y = \text{Sin}(x)$, где:

y – имя переменной, которой будет присвоено вычисленное значение; Sin – имя вызываемой функции; x – выражение, являющееся аргументом функции.

Примеры вызовов функции sin в Паскале:

a := sin (b);

a := sin (t + 3.14*b);

a := sin (a + abs(sin (b)));

Понятно, что заранее заданный набор стандартных функций может быть недостаточным для нужд программиста, поэтому практически все языки программирования предоставляют ему возможность самому определять собственные дополнительные функции и использовать их в программе. Выше приведены примеры описания функции с именем F, вычисляющей значение выражения

$F = 4*(x - 1)*(x - 3)$, где x – аргумент. А вот примеры её использования в основной программе:

Бейсик	Паскаль
R = F(T)	R:=F(t);
Си	Алгоритмический
R = F(t);	R:= F(t)

Видно, что вызов функции F аналогичен вызову стандартных функций.

В алгоритмах этого задания присутствуют исходные данные, но ручное выполнение алгоритма весьма трудоемко, поэтому, как и в предыдущей задаче, нужно понять, в чем цель алгоритма.

21 (В14 демоверсии 2012 г.)

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F (x) F = 4 * (x - 1) * (x - 3) END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 4*(x-1)*(x-3); end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R)then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический язык
<pre> int F(int x) { return 4*(x-1)*(x-3); } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел a, b, t, M, R a := -20; b := 20 M := a; R:= F(a) нц для t от a до b если F(t)< R то M := t; R := F(t) все кц вывод M кон алг цел F(цел x) нач знач := 4*(x-1)*(x-3) кон </pre>

Решение:

Для того, чтобы определить назначение выводимой на печать переменной M , сначала изучим текст главной программы. Её основным элементом является цикл

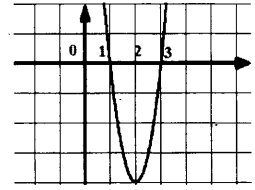
```

for t := a to b do
    begin
        if (F(t)<R)then begin
            M := t;
            R := F(t);
        end;
    end;
end;
        
```

В этом цикле ищется минимум функции $F(t)$ на множестве целых значений t от a до b с единичным шагом. Значение t , на котором функция достигает своего минимума, запоминается в переменной M .

Итак, M – значение, на котором функция $F(x) = 4*(x-1)*(x-3)$ достигает минимума на множестве целых x , таких что $-20 \leq x \leq 20$.

Функция $F(x) = 4*(x-1)*(x-3)$ является квадратным трехчленом с корнями 1 и 3. Задача свелась к нахождению абсциссы минимума функции на заданном множестве целых чисел. Это можно сделать самыми различными способами, например, построив эскиз графика функции. Графиком данного квадратного трехчлена будет парабола, симметричная относительно оси $x=2$ (среднее арифметическое её корней), ветви которой направлены вверх. Поэтому экстремум функции достигается при $x=2$, и это будет именно минимум. Найденное значение x целое и удовлетворяет условию $-20 \leq x \leq 20$.



Ответ: 2.

21 Пример (ОС 2014 г.)

Напишите в ответе число различных значений входной переменной k , при которых программа выдаёт тот же ответ, что и при входном значении $k = 64$. Значение $k = 64$ также включается в подсчёт различных значений k . Для Вашего удобства программа приведена на четырёх языках программирования.

Бейсик	Паскаль
<pre> DIM K, I AS LONG INPUT K I = 12 WHILE I > 0 AND F(I) > K I = I - 1 WEND PRINT I FUNCTION F(N) F = N * N * N END FUNCTION </pre>	<pre> var k, i : longint; function f(n: longint) : longint; begin f := n * n * n end; begin readln(k); i := 12; while (i>0) and (f(i)>k) do i := i-1; writeln(i) end. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> long f(long n) { return n * n * n; } void main() { long k, i; scanf("%ld", &k); i = 12; while (i>0 && f(i)>k) do i- -; printf("%ld", i); } </pre>	<pre> алг нач цел i, k ввод k i := 12 нц пока i > 0 и f(i) > k i := i - 1 кц вывод i кон алг цел f(цел n) нач знач := n * n * n кон </pre>

Решение:

При решении данной задачи следует выполнить два действия:

Во-первых, определить, что за функция вызывается в основном алгоритме. Для данной задачи это, как хорошо видно из описания, куб числа ($f=n^3$).

Во-вторых, определить, при каком условии завершается выполнение цикла. Для любого целого $k > 0$ цикл завершается в ситуации, когда $f(i) \leq k$ (при $k > 0$ это наступает раньше, чем $i \leq 0$). Следует, таким образом, определить, при каком i выражение $i^3 \leq 64$ истинно. Очевидно, что это все $i \leq 4$.

Итак, при $k=64$ программа выведет число 4 (значение переменной i , такое, что $i^3 = 64$). При $9 \leq k < 64$ программа выведет число 3, при $k=125$ программа выведет число 5. Таким образом, число 4 будет выводиться программой для всех $64 \leq k < 125$. Таких целых чисел в этом диапазоне ровно 61.

Ответ: 61.

Задачи для самостоятельного решения

21.1

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F (x) F = 10 * (x+2) * (x-6) END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 10*(x+2)*(x-6); end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический язык
<pre> int F(int x) { return 10*(x+2)*(x-6); } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел a, b, t, M, R a := -20; b := 20 M := a; R := F(a) нц для t от a до b если F(t) < R то M := t; R := F(t) все кц вывод M кон алг цел F(цел x) нач знач := 10*(x+2)*(x-6) кон </pre>

21.2

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F (x) F = -2 * (x-2)*(x-6) END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := -2*(x-2)*(x-6); end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)>R)then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический язык
<pre> int F(int x) { return -2*(x-2)*(x-6); } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> a, b, t, M, R a := -20; b := 20 M := a; R:= F(a) <u>нц для</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t)> R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u> M <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> := -2*(x-2)*(x-6) <u>кон</u> </pre>

21.3

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -30: B = 40 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F (x) F = ABS(x-2)+1 END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := abs(x-2)+1; end; BEGIN a := -30; b := 40; M := a; R := F(a); for t := a to b do begin if (F(t)<R)then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический язык
<pre> int F(int x) { return abs(x-2)+1; } void main() { int a, b, t, M, R; a = -30; b = 40; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел a, b, t, M, R a := -30; b := 40 M := a; R:= F(a) нц для t от a до b если F(t)< R то M := t; R := F(t) все кц вывод M кон алг цел F(цел x) нач знач := iabs(x-2)+1 кон </pre>

21.4

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = 5: B = 25 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F (x) F = 3 * (x+2) * (x-6) END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 3*(x+2)*(x-6); end; BEGIN a := 5; b := 25; M := a; R := F(a); for t := a to b do begin if (F(t)<R)then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический язык
<pre> int F(int x) { return 3*(x+2)*(x-6); } void main() { int a, b, t, M, R; a = 5; b = 25; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> a, b, t, M, R a := 5; b := 25 M := a; R:= F(a) <u>нц для</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t)< R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u> M <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> := 3*(x+2)*(x-6) <u>кон</u> </pre>

21.5

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F (x) F = -2 * (x-1)*(x-1) END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := -2*(x-1)*(x-1); end; BEGIN a := -10; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)>R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический язык
<pre> int F(int x) { return -2*(x-1)*(x-1); } void main() { int a, b, t, M, R; a = -10; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> a, b, t, M, R a := -10; b := 20 M := a; R:= F(a) <u>нц для</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t) > R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u> M <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> := -2*(x-1)*(x-1) <u>кон</u> </pre>

21.6

Напишите в ответе число различных значений входной переменной k , при которых программа выдаёт тот же ответ, что и при входном значении $k = 120$. Значение $k = 120$ также включается в подсчёт различных значений k . Для Вашего удобства программа приведена на четырёх языках программирования.

Бейсик	Паскаль
<pre> DIM K, I AS LONG INPUT K I = 12 WHILE I > 0 AND F(I) > K I = I - 1 WEND PRINT I FUNCTION F(N) IF N < 2 THEN F = 1 ELSE F = N * F(N-1) END IF END FUNCTION </pre>	<pre> var k, i : longint; function f(n: longint): longint; begin if n<2 then f := 1 else f := n*f(n-1) end; begin readln(k); i := 12; while (i>0) and (f(i)>k) do i := i-1; writeln(i) end. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> long f(long n) { if (n<2) return 1; else return n*f(n-1); } void main() { long k, i; scanf("%ld", &k); i = 12; while (i>0 && f(i)>k) do i- -; printf("%ld", i); } </pre>	<pre> алг нач цел i, k ввoд k i := 12 нц пока i > 0 и f(i) > k i := i-1 кц вывод i кон алг цел f(цел n) нач если n < 2 то знач := 1 иначе знач := n*f(n-1) все кон </pre>

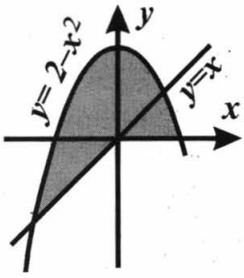
2.3. Задания по программированию с развернутым ответом

Среди четырех заданий с развернутым ответом три: 24(C1), 25(C2) и 27(C4) – проверяют умение читать, модифицировать, создавать записывать алгоритмы на языке программирования. Рассмотрим их последовательно.

Задание 24 (C1)

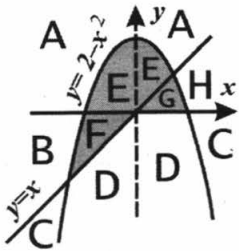
Задание C1 в демоверсии 2012г. незначительно отличалось от заданий прошлых лет. Разница заключается в том, что теперь от учащегося требовалось не привести пример входных данных, когда программа работает неправильно, а заполнить табличку правильности работы программы, столбцы которой содержат условия, а строки – возможные области исходных данных.

C1 (демоверсия 2012)



Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.

Бейсик	Паскаль
<pre> INPUT x, y IF y>=x THEN IF y>=0 THEN IF y<=2-x*x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END </pre>	<pre> var x,y: real; begin readln(x,y); if y>=x then if y>=0 then if y<=2-x*x then write('принадлежит') else write('не принадлежит') end. </pre>
Си	Алгоритмический язык
<pre> void main(void){ float x,y; scanf("% f % f",&x,&y); if (y>=x) if (y>=0) if (y<=2-x*x) printf("принадлежит"); else printf("не принадлежит"); } </pre>	<pre> алг нач вещ x, y ввод x, y если y>=x то если y>=0 то если y<=2-x*x то вывод 'принадлежит' иначе вывод 'не принадлежит' все все все кон </pre>



Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G и H). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($y \geq x$)	Условие 2 ($y \geq 0$)	Условие 3 ($y \leq 2 - x^2$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "-" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "-" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

Решение:

Заполнив таблицу для каждой области, получим:

Область	Условие 1 ($y >= x$)	Условие 2 ($y >= 0$)	Условие 3 ($y <= 2 - x * x$)	Программа выведет	Область обрабатывается верно
A	да	да	нет	не принадлежит	да
B	да	нет	-	-	нет
C	нет	-	-	-	нет
D	нет	-	-	-	нет
E	да	да	да	принадлежит	да
F	да	нет	-	-	нет
G	нет	-	-	-	нет
H	нет	-	-	-	нет

Как и демоверсиях прошлых лет, в этой задаче типичными ошибками программиста, подлежащими исправлению, являются следующие:

- 1) неправильно расставлены операторные скобки в операторе ветвления
- 2) отсутствует одно или несколько необходимых условий

В данном случае, самым простым способом исправления обеих ошибок является замена ошибочного оператора ветвления (пример для Паскаля)

```

if y >= x then
  if y >= 0 then
    if y <= 2 - x * x then
      write('принадлежит')
    else
      write('не принадлежит')

```

на правильный, например

```

if (x < 0) and (y <= 2 - x * x) and (y >= x) or (x >= 0) and (y >= 0) and (y <= 2 - x * x)
then
  write('принадлежит')
else
  write('не принадлежит')

```

В 2014 г. задание C1 претерпело значительные изменения и формулируется примерно следующим образом.

24 (C1 ОС 2014)

На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество чётных чисел в исходной последовательности и минимальное чётное число. Если чётных чисел нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают 1000. Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на четырёх языках программирования.

Бейсик	Паскаль
<pre> CONST n = 4 count = 0 minimum = 0 FOR I = 1 TO n INPUT x IF x mod 2 = 0 THEN count = count + 1 IF x < minimum THEN minimum = I END IF END IF NEXT I IF count > 0 THEN PRINT count PRINT minimum ELSE PRINT "NO" END IF </pre>	<pre> const n = 4; var i, x: integer; var minimum, count: integer; begin count := 0; minimum := 0; for i := 1 to n do begin read(x); if x mod 2 = 0 then begin count := count + 1; if x < minimum then minimum := i end end; if count > 0 then begin writeln(count); writeln(minimum) end else writeln('NO') end; end; end. </pre>
Си	Алгоритмический язык
<pre> #include <stdio.h> #define n 4 void main(void) { int i, x; int minimum, count; count = 0; minimum = 0; for (i = 1; i <= n; i++) { scanf("%d",&x); if (x % 2 == 0) { count++; if (x < minimum) minimum = i; } } if (count > 0) { printf("%d\n", count); printf("%d\n", minimum); } else printf("NO\n"); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> n = 4 <u>цел</u> i, x <u>цел</u> minimum, count count := 0 minimum := 0 <u>нц для i от 1 до n</u> <u>ввод</u> x <u>если</u> mod(x, 2) = 0 <u>то</u> count := count + 1 <u>если</u> x < minimum <u>то</u> minimum := i <u>все</u> <u>все</u> <u>кц</u> <u>если</u> count > 0 <u>то</u> <u>вывод</u> count, <u>нс</u> <u>вывод</u> minimum <u>иначе</u> <u>вывод</u> "NO" <u>все</u> <u>кон</u> </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности:

2 9 4 3

2. Приведите пример такой последовательности, содержащей хотя бы одно чётное число, что, несмотря на ошибки, программа печатает правильный ответ.

3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Примечание: 0 – чётное число.

Решение:

Решение использует запись программы на Паскале.

1. Программа выведет два числа: 2 и 0.

2. Пример последовательности, содержащей чётные числа, для которой программа работает правильно: 1 3 5 0.

Объяснение. В конце работы программы значение переменной `minimum` всегда равно 0. Соответственно, программа будет работать верно, если в последовательности есть 0. Выведенное количество чётных чисел будет правильным в любом случае.

3. В программе есть две ошибки.

Первая ошибка: неверная инициализация `minimum`.

Строка с ошибкой:

```
minimum := 0;
```

Верное исправление:

```
minimum := 1001;
```

Вместо 1001 может быть любое целое число, большее 1000, либо `MAXINT`.

Можно использовать и число 1000, так как при выводе мы проверяем, есть ли в последовательности хотя бы одно чётное число.

Вторая ошибка: неверное присваивание при вычислении минимума.

Строка с ошибкой:

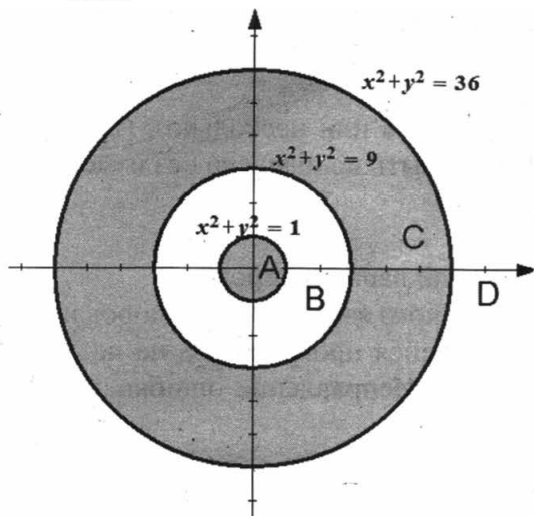
```
minimum = i;
```

Верное исправление:

```
minimum = x;
```


Задачи для самостоятельного решения

C1.1



Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.

Бейсик	Паскаль
<pre> INPUT x, y IF x*x + y*y <=36 THEN IF x*x + y*y >=9 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END </pre>	<pre> var x,y: real; begin readln(x,y); if x*x + y*y <=36 then if x*x + y*y >=9 then write('принадлежит') else write('не принадлежит') end. </pre>
Си	Алгоритмический язык
<pre> void main(void){ float x,y; scanf("%f%f",&x,&y); if (x*x + y*y <=36) if (x*x + y*y >=9) printf("принадлежит"); else printf("не принадлежит"); } </pre>	<pre> алг нач вещь x, y ввод x, y если x*x + y*y <=36 то если x*x + y*y >=9 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все кон </pre>

Последовательно выполните следующее.

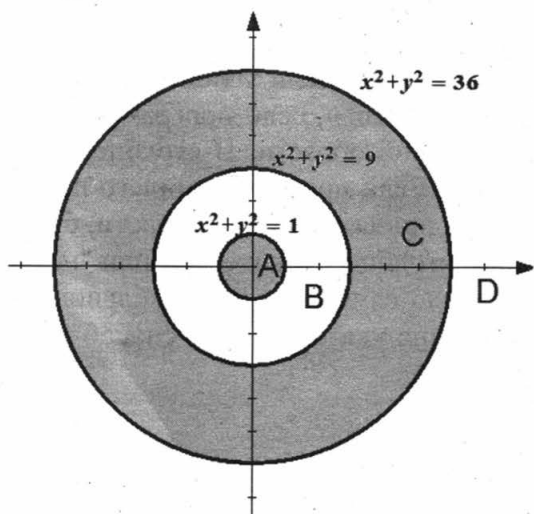
1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D).

Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 $x^2+y^2 \leq 36$	Условие 2 $x^2+y^2 \geq 9$	Программа выведет	Область обрабатывается верно
A				
B				
C				
D				

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "-" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "-" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".
 2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C1.2



Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.

<p>Бейсик</p> <pre> INPUT x, y IF x*x + y*y <= 1 THEN IF x*x + y*y >=9 THEN IF x*x + y*y <= 36 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF ENDIF END </pre>	<p>Паскаль</p> <pre> var x,y: real; begin readln(x,y); if x*x + y*y <= 1 then if x*x + y*y >= 9 then if x*x + y*y <= 36 then write('принадлежит') else write('не принадлежит') end. end. </pre>
<p>Си</p> <pre> void main(void){ float x,y; scanf("%f%f",&x,&y); if (x*x + y*y <= 1) if (x*x + y*y >=9) if (x*x + y*y <= 36) printf("принадлежит"); else printf("не принадлежит"); } } </pre>	<p>Алгоритмический язык</p> <pre> алг нач вещ x, y ввод x, y если x*x + y*y <=1 то если x*x + y*y >=9 то если x*x + y*y <=36 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все все кон </pre>

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D).

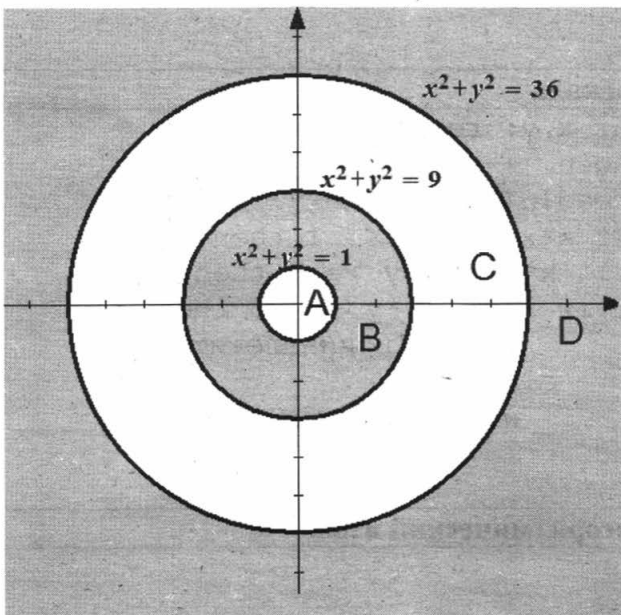
Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 $x^2 + y^2 \leq 1$	Условие 2 $x^2 + y^2 \geq 9$	Условие 3 $x^2 + y^2 \leq 36$	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "-" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "-" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C1.3



Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.

Бейсик	Паскаль
<pre> INPUT x, y IF x*x + y*y >= 36 THEN IF x*x + y*y >= 1 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END </pre>	<pre> var x, y: real; begin readln(x, y); if x*x + y*y >= 36 then if x*x + y*y >= 1 then write('принадлежит') else write('не принадлежит') end. </pre>

Си	Алгоритмический язык
<pre>void main(void) { float x,y; scanf ("%f%f", &x, &y); if (x*x + y*y >=36) if (x*x + y*y >=1) printf("принадлежит"); else printf("не принадлежит"); }</pre>	<pre>алг нач вещ x, y ввод x, y если x*x + y*y >=36 то если x*x + y*y >=1 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все кон</pre>

Последовательно выполните следующее.

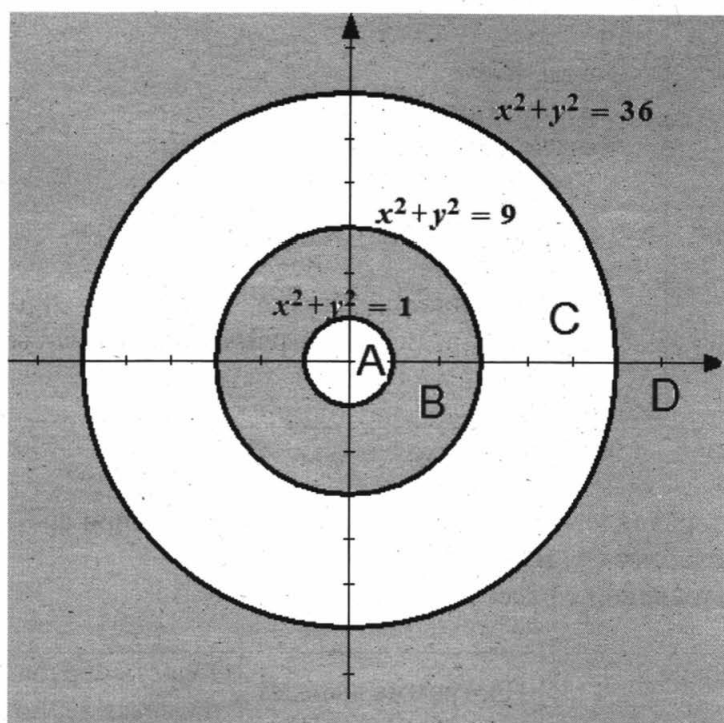
1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D).

Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 $x^2+y^2 \geq 36$	Условие 2 $x^2+y^2 \geq 1$	Программа выведет	Область обрабатывается верно
А				
В				
С				
D				

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "-" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "-" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)



Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.

Бейсик	Паскаль
<pre> INPUT x, y IF x*x + y*y <=9 THEN IF x*x + y*y >=1 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END </pre>	<pre> var x,y: real; begin readln(x,y); if x*x + y*y <= 9 then if x*x + y*y >= 1 then write('принадлежит') else write('не принадлежит') end. </pre>
Си	Алгоритмический язык
<pre> void main(void){ float x,y; scanf("%f%f",&x,&y); if (x*x + y*y <=9) if (x*x + y*y >=1) printf("принадлежит"); else printf("не принадлежит"); } </pre>	<pre> алг нач вещ x,y ввод x,y если x*x + y*y <=9 то если x*x + y*y >=1 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все кон </pre>

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D).

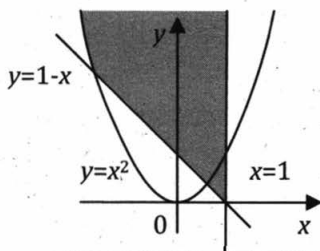
Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 $x*x+y*y \leq 9$	Условие 2 $x*x+y*y \geq 1$	Программа выведет	Область обрабатывается верно
A				
B				
C				
D				

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "-" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "-" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

Полезно также повторить задания С1 прошлых лет.

C1.5 (OC)



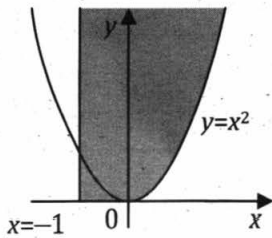
Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы), неограниченной сверху. Программист торопился и написал программу неправильно.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var x,y: real; begin readln(x,y); if x<=1 then if y>=1-x then if (y>=x*x) then write('принадлежит') else write('не принадлежит') end.</pre>	<pre>INPUT x, y IF x<=1 THEN IF y>=1-x THEN IF y>=x*x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void){ float x,y; scanf("%f%f",&x,&y); if (x<=1) if (y>=1-x) if (y>=x*x) printf("принадлежит"); else printf("не принадлежит") }</pre>

Последовательно выполните следующее.

1. Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C1.6 (OC)



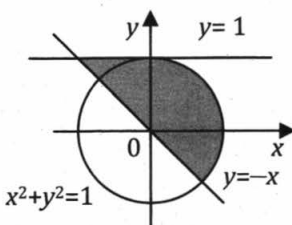
Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы), неограниченной сверху. Программист торопился и написал программу неправильно.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var x,y: real; begin readln(x,y); if y>=0 then if x>=-1 then if (y>=x*x) then write('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x, y IF y>=0 THEN IF x>=-1 THEN IF y>=x*x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void){ float x,y; scanf("%f%f",&x,&y); if (y>=0) if (x>=-1) if (y>=x*x) printf("принадлежит"); else printf("не принадлежит"); }</pre>

Последовательно выполните следующее.

1. Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C1.7 (OC)



Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var x,y: real; begin readln(x,y); if y>=-x then if y<=1 then if (x*x+y*y<=1) then write('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x, y IF y>=-x THEN IF y<=1 THEN IF x*x+y*y<=1 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void){ float x,y; scanf("%f%f",&x,&y); if (y>=-x) if (y<=1) if (x*x+y*y<=1) printf("принадлежит"); else printf("не принадлежит"); }</pre>

Последовательно выполните следующее.

1. Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C1.8 (ОС 2014)

На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество чётных чисел в исходной последовательности и сумму таких чисел. Если чётных чисел нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают 1000. Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на четырёх языках программирования.

Бейсик	Паскаль
<pre>CONST n = 4 count = 0 sum = 0 FOR I = 1 TO n INPUT x IF x mod 2 = 0 THEN count = count + 1 sum = x END IF NEXT I IF sum > 0 THEN PRINT count PRINT sum ELSE PRINT "NO" ENDIF</pre>	<pre>const n = 4; var i, x: integer; var sum, count: integer; begin count := 0; sum := 0; for i := 1 to n do begin read(x); if x mod 2 = 0 then begin count := count + 1; sum := x end end; if sum > 0 then begin writeln(count); writeln(sum) end else writeln('NO') end. end.</pre>

Си	Алгоритмический язык
<pre> #include <stdio.h> #define n 4 void main(void) { int i, x; int sum, count; count = 0; sum = 0; for (i = 1; i <= n; i++) { scanf("%d", &x); if (x % 2 == 0) { count++; sum = x; } } if (sum > 0) { printf("%d\n", count); printf("%d\n", sum); } else printf("NO\n"); } </pre>	<pre> алг нач цел n = 4 цел i, x цел sum, count count := 0 sum := 0 нц для i от 1 до n ввод x если mod(x, 2) = 0 то count := count + 1 sum := x все кц если sum > 0 то вывод count, нс вывод sum, нс иначе вывод "NO" все кон </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности:

20, 93, 40, 39

2. Приведите пример такой последовательности, содержащей хотя бы одно чётное число, что, несмотря на ошибки, программа печатает правильный ответ.

3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Примечание: 0 – чётное число.

С1.9 (ОС 2014 г.)

На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество чётных чисел в исходной последовательности и максимальное чётное число. Если чётных чисел нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают 1000. Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на четырёх языках программирования.

Бейсик	Паскаль
<pre> CONST n = 4 count = 0 maximum = 1000 FOR I = 1 TO n INPUT x IF x mod 2 = 0 THEN count = count + 1 IF x < maximum THEN maximum = x END IF END IF NEXT I IF count > 0 THEN PRINT count PRINT maximum ELSE PRINT "NO" END IF </pre>	<pre> const n = 4; var i, x: integer; var maximum, count: integer; begin count := 0; maximum := 1000; for i := 1 to n do begin read(x); if x mod 2 = 0 then begin count := count + 1; if x < maximum then maximum := x end end; if count > 0 then begin writeln(count); writeln(maximum) end else writeln('NO') end end end. </pre>
Си	Алгоритмический язык
<pre> #include <stdio.h> #define n 4 void main(void) { int i, x; int maximum, count; count = 0; maximum = 1000; for (i = 1; i <= n; i++) { scanf("%d",&x); if (x % 2 == 0) { count++; if (x < maximum) maximum = x; } } if (count > 0) { printf("%d\n", count); printf("%d\n", maximum); } else printf("NO\n"); } </pre>	<pre> алг нач цел n = 4 цел i, x цел maximum, count count := 0 maximum := 1000 нц для i от 1 до n ввод x если mod(x, 2) = 0 то count := count + 1 если x < maximum то maximum := x все все кц если count > 0 то вывод count, нс вывод maximum иначе вывод "NO" все кон </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности:

6 9 4 3

2. Приведите пример такой последовательности, содержащей хотя бы одно чётное число, что, несмотря на ошибки, программа печатает правильный ответ.

3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Примечание: 0 – чётное число.

Задание 25 (С2)

Задание С2 демоверсии 2014 г. не имело существенных отличий от заданий прошлых лет. В задании С2 допускается словесное описание алгоритма т.е. это задание можно выполнить, не владея ни одним языком программирования. Тем не менее, практика показывает, что подавляющее большинство учащихся, приступивших к этому заданию, выполняли его на одном из языков программирования. Именно поэтому это задание обсуждается в данном разделе книги. Для выполнения задания С2 необходимо уметь для непустого множества элементов в массиве, определяемого некоторым условием (числа, бóльшие или меньшие определенного значения, делящиеся на заданное число и т.д.) вычислять определенную величину – максимальный (минимальный) элемент этого множества, их сумму, среднее арифметическое и т.д.

С2 (демоверсия 2012 г.)

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести минимальное значение среди элементов массива, которые имеют чётное значение и не делятся на три. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого чётно и не кратно трем.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но использовать все описанные переменные не обязательно.

Паскаль	Алгоритмический язык
<pre> const N = 20; var a: array [1..N] of integer; i, j, min: integer; begin for i := 1 to N do readln(a[i]); ... end.</pre>	<pre> алг нач цел N = 20 целтаб a[1:N] цел i, j, MIN нц для i от 1 до N ввод a[i] кц ... кон</pre>
Бейсик	СИ
<pre> N = 20 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre> #include <stdio.h> #define N 20 void main(void){ int a[N]; int i, j, min; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>

Русский (естественный) язык

Объявляем массив A из 20 элементов.
Объявляем целочисленные переменные I, J, MIN.
В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й.

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Решение:

Для проверки делимости числа на 2 или 3 будем использовать операцию получения остатка от деления. Изначально переменной min присваиваем значение, заведомо большее любого из элементов массива. Далее ищем минимум среди элементов, отвечающих заданному условию.

На языке Паскаль

```
min := 1000;
for i:=1 to N do
    if (a[i] mod 2=0) and (a[i] mod 3<>0) and (a[i]<min) then
        min := a[i];
writeln(min);
```

На Алгоритмическом языке

```
MIN := 1000;
нц для i от 1 до N
    если mod(a[i],2)=0 и mod(a[i],3)<>0 и a[i]<MIN
        то
            MIN := a[i]
    все
кц
вывод MIN
```

На языке Бейсик

```
MIN = 1000
FOR I = 1 TO N
    IF (A(I) MOD 2 = 0) AND (A(I) MOD 3 <> 0) AND (A(I) < MIN) THEN
        MIN = A(I)
    END IF
NEXT I
PRINT MIN
```

На языке СИ

```
min = 1000;
for (i=0; i<N; i++)
    if (a[i]%2==0 && a[i]%3!=0 && a[i]<min)
        min = a[i];
printf("%d", min);
```

На естественном языке

Записываем в переменную MIN начальное значение, равное 1000. В цикле от первого элемента до двадцатого находим остаток от деления элемента исходного массива на два и на три. Если остаток от деления на два равен нулю и остаток от деления на три не равен нулю, то сравниваем значение текущего элемента массива со значением переменной MIN. Если текущий элемент массива меньше MIN, то записываем в MIN значение этого элемента массива. Переходим к следующему элементу.

После завершения цикла выводим значение переменной MIN.

Задачи для самостоятельного решения (ОС)

С2.1

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от -1000 до 1000 . Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет найти и вывести максимальное значение среди положительных элементов массива, кратных пяти. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого положительно и делится на пять.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Бейсик
<pre> const N=20; var a: array [1..N] of integer; i, j, max: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre> N=20 DIM A(N) AS INTEGER DIM I, J, MAX AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
СИ	Естественный язык
<pre> #include <stdio.h> #define N 20 void main(void) { int a[N]; int i, j, max; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<p>Объявляем массив А из 20 элементов. Объявляем целочисленные переменные I, J, MAX. В цикле от 1 до 20 вводим элементы массива А с 1-го по 20-й.</p> <p>...</p>

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

C2.2

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от -1000 до 1000 . Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести минимальное значение среди элементов массива, которые имеют чётное значение и не делятся на пять. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого чётно и не кратно пяти.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Бейсик
<pre> const N=20; var a: array [1..N] of integer; i, j, min: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre> N=20 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>

СИ	Естественный язык
<pre>#include <stdio.h> #define N 20 void main(void) { int a[N]; int i, j, min; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<p>Объявляем массив А из 20 элементов. Объявляем целочисленные переменные I, J, MIN. В цикле от 1 до 20 вводим элементы массива А с 1-го по 20-й.</p> <p>...</p>

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

C2.3

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 10 – баллы учащихся за тест по информатике. Для получения положительной оценки за тест требовалось набрать не менее 4 баллов. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет найти и вывести минимальный балл среди учащихся, получивших за тест положительную оценку. Известно, что в классе хотя бы один учащийся получил за тест положительную оценку.

Паскаль	Бейсик
<pre>const N=20; var a: array [1..N] of integer; i, j, min: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=20 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
СИ	Естественный язык
<pre>#include <stdio.h> #define N 20 void main(void) {int a[N]; int i, j, min; for (i=0; i<N; i++) scanf("% d", &a[i]); ... }</pre>	<p>Объявляем массив А из 20 элементов. Объявляем целочисленные переменные I, J, MIN. В цикле от 1 до 20 вводим элементы массива А с 1-го по 20-й.</p> <p>...</p>

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

C2.4

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 100 – баллы учащихся выпускного класса за итоговый тест по информатике. Для получения положительной оценки за тест требовалось набрать не менее 20 баллов. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет найти и вывести минимальный балл среди учащихся, получивших за тест положительную оценку. Известно, что в классе хотя бы один учащийся получил за тест положительную оценку.

Паскаль	Бейсик
<pre>const N=30; var a: array [1..N] of integer; i, j, min: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=30 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
СИ	Естественный язык
<pre>#include <stdio.h> #define N 30 void main(void) {int a[N]; int i, j, min; for (i=0; i<N; i++) scanf("% d", &a[i]); ... }</pre>	<p>Объявляем массив А из 30 элементов. Объявляем целочисленные переменные I, J, MIN. В цикле от 1 до 30 вводим элементы массива А с 1-го по 30-й. ...</p>

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

C2.5

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от –10 000 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, сумма которых нечётна и отрицательна. Под парой подразумевается два подряд идущих элемента массива.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик	Паскаль
<pre>N = 20 DIM A(N) AS INTEGER DIM I, J, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>const N = 20; var a: array [1..N] of integer; i, j, k: integer; begin for i := 1 to N do readln(a[i]); ... end.</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> #define N 20 void main() { int a[N]; int i, j, k; for (i = 0; i < N; i++) scanf("%d", &a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N = 20 <u>цел таб</u> a[1:N] <u>цел</u> i, j, k <u>нц для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>
Естественный язык	
<p>Объявляем массив А из 20 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 20 вводим элементы массива А с 1-го по 20-й. ...</p>	

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6) или в виде блок-схемы. В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Задание 27 (С4)

Задание С4 нацелено на проверку умения создавать собственные программы (30–50 строк) для решения задач средней сложности. Это задание отличается от других заданий ЕГЭ по информатике максимальным первичным баллом – 4 балла и наибольшим запланированным временем – 55 минут из 235 минут экзамена. В этом задании проверяется не только умение составить алгоритм, но и написать законченную программу на одном из языков программирования (по выбору экзаменуемого), т.е. владение технологией программирования.

Для выполнения С4 необходимо свободное владение подмножеством языка программирования, включающим описание типов данных, работу с массивами, операции с числовыми, логическими, символьными и строковыми данными, ввод-вывод данных, организацию ветвлений и циклов. Помимо этого желательно знать и уметь правильно применять алгоритмы, перечисленные в Кодификаторе элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для проведения ЕГЭ по информатике, а именно:

- Нахождение минимального (максимального) значения в данном массиве и количества элементов, равных ему, за однократный просмотр массива.
- Нахождение второго по величине (второго максимального или второго минимального) значения в данном массиве за однократный просмотр массива.
- Сортировка массива.
- Слияние двух упорядоченных массивов в один без использования сортировки.
- Обработка отдельных символов данной строки. Подсчет частоты появления символа в строке.
- Работа с подстроками данной строки с разбиением на слова по пробельным символам. Поиск подстроки внутри данной строки, замена найденной подстроки на другую строку.

Задание С4 рекомендуется выполнять поэтапно:

- 1) Внимательный анализ условия задачи. Уяснение формата входных и выходных данных, сути преобразований, которые требуется выполнить над входными данными для получения нужного результата.
- 2) Выбор представления и структуры данных программы, проектирование алгоритма решения задачи
- 3) Написание текста программы
- 4) Проверка написанной программы на наличие синтаксических и логических ошибок. Тестирование программы в «ручном режиме» с уделением особого внимания тестам для граничных значений исходных данных.

С4 (демоверсия 2012 г.)

В командных олимпиадах по программированию для решения предлагается не больше 11 задач. Команда может решать предложенные задачи в любом порядке. Подготовленные решения команда посылает в единую проверяющую систему соревнований. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать пришедшие запросы, чтобы определить наиболее популярные задачи. Следует учитывать, что количество запросов в списке может быть очень велико, так как многие соревнования проходят с использованием Интернет.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подаётся количество пришедших запросов N . В каждой из последующих N строк записано название задачи в виде текстовой строки. Длина строки не превосходит 100 символов, название может содержать буквы, цифры, пробелы и знаки препинания.

Пример входных данных:

```
6
A+B
Крестики-Нолики
Прямоугольник
Простой делитель
A+B
Простой делитель
```

Программа должна вывести список из трёх наиболее популярных задач с указанием количества запросов по ним. Если в запросах упоминаются менее трех задач, то выведите информацию об имеющихся задачах. Если несколько задач имеют ту же частоту встречаемости, что и третья по частоте встречаемости задача, их тоже нужно вывести.

Пример выходных данных для приведённого выше примера входных данных:

A+B 2

Простой делитель 2

Крестики-Нолики 1

Прямоугольник 1

Решение:

Программа читает все входные данные один раз, не запоминая их в массиве, размер которого равен N, а составляя только список встретившихся задач и количества запросов по каждой из них. Во время чтения данных об очередной задаче просматривается список ранее сохраненных задач; если она уже есть в списке, то количество запросов по ней увеличивается на 1, иначе задача добавляется в массив упомянутых в запросах задач (при корректных данных он не может быть больше 11). После окончания ввода производится сортировка массивов задач и количества запросов, отданных за них, в порядке убывания количества запросов, затем выводится список из трёх первых задач с указанием частоты встречаемости (или весь список, если его длина меньше трёх). Вместо сортировки можно применить и алгоритм поиска трёх максимальных элементов в массиве. Затем выводятся задачи, частота встречаемости которых не ниже, чем у третьей задачи. Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая. Ниже приведены примеры решения задания на Алгоритмическом языке, а также на языках Паскаль и Бейсик. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования. Так, на языке C++ при считывании строковой переменной будет считано не все название задачи, а только его первое слово, поэтому следует использовать функцию `getline(cin, s)`, аналогичная проблема возникает и в языке Си.

Пример правильной и эффективной программы на языке Паскаль:

```
Var N, Num, i, j, t: integer;
    Count: array[1..11] of integer;
    s: string;
    Names: array[1..11] of string;
Begin
    Num:=0; {Число различных задач в списке запросов}
    ReadLn(N); {Считываем количество запросов}
    for i:=1 to N do
    begin
        ReadLn(s); {считали очередную задачу}
        {Осуществляем ее поиск в списке уже встретившихся}
        j:=1;
        while (j<=Num) and (s<>Names[j]) do j:=j+1;
        {Если она найдена}
        if j<=Num then {Увеличиваем счетчик числа запросов}
            Count[j]:=Count[j]+1
        else begin {Иначе добавляем задачу в конец списка}
            Names[j]:=s;
            Count[j]:=1;
            Num:=Num+1
        end
    end;
    {Сортируем массивы Names и Count в порядке убывания значений массива Count}
    for i:=Num downto 2 do
    for j:=2 to i do if Count[j-1]<Count[j] then
    begin
```

```

t:=Count[j]; Count[j]:=Count[j-1]; Count[j-1]:=t;
s:=Names[j]; Names[j]:=Names[j-1]; Names[j-1]:=s;
end;
if Num >= 3 then j := 3 else j := Num;
i := 1;
while (i <= Num) and (Count[i] >= Count[j]) do
begin
  WriteLn(Names[i], ' ', Count[i]);
  i := i + 1;
end
end.

```

Пример правильной и эффективной программы на Алгоритмическом языке:

```

литтаб Names[1:11] | названия задач
целтаб Count[1:11] | счетчики числа запросов по каждой задаче
цел i, j, t
лит s
| 1. Чтение списка запросов
| 1.1. Инициализация количества запросов и счетчика задач
Num:=0 | Число различных задач в списке запросов
ввод N | Считываем количество запросов
| 1.2. Цикл чтения.
нц для i от 1 до N
  ввод s | Считали очередную задачу
          | Осуществляем ее поиск в списке уже встретившихся
  j:=1
  нц пока (j<=Num) и (s<>Names[j])
    j:=j+1
  кц
  | Обработываем очередную задачу
если j<=Num | Если задача найдена в списке
  то | Увеличиваем счетчик числа запросов
    Count[j]:=Count[j]+1
  иначе | Добавляем задачу в конец списка
    Names[j]:=s
    Count[j]:=1
    Num:=Num+1
все
кц
| 2. Совместно сортируем массивы Names и Count
| в порядке убывания значений массива Count
нц для i от Num до 2 шаг -1
  нц для j от 2 до i
    если Count[j-1]<Count[j] то
      t:=Count[j]; Count[j]:=Count[j-1]; Count[j-1]:=t
      s:=Names[j]; Names[j]:=Names[j-1]; Names[j-1]:=s
    все
  кц
кц
| 3. Вывод задач-"призеров"
| 3.1. Определение порога для количества запросов по задаче
| Порог равен Count[j]
если Num >= 3
  то j := 3

```

```

    иначе j := Num;
всё
|      3.2. Цикл вывода
i := 1;
нц пока (i <= Num) и (Count[i] >= Count[j])
    вывод нс, Names[i], ' ', Count[i]
    i := i + 1
кц
Кон

```

Пример правильной и эффективной программы на языке Бейсик:

```

DIM N, Num, i, j, t AS INTEGER
DIM Count(11) AS INTEGER
DIM s AS STRING
DIM Names(11) AS STRING
REM Число различных задач в списке запросов
Num = 0
REM считываем количество запросов
INPUT N
FOR i = 1 TO N
REM считываем очередную задачу
    INPUT s
REM осуществляем ее поиск в списке уже встретившихся
    j = 1
    WHILE j <= Num AND s <> Names(j)
        j = j + 1
    WEND
    IF j <= Num THEN
REM Если она найдена, увеличиваем счетчик числа запросов
        Count(j) = Count(j)+1
    ELSE
REM Иначе добавляем задачу в конец списка
        Names(j) = s: Count(j) = 1
        Num = Num + 1
    ENDIF
NEXT i
REM Сортируем массивы Names и Count
REM в порядке убывания значений массива Count
FOR i = Num TO 2 Step -1
    FOR j = 2 TO i
        IF Count(j-1) < Count(j) THEN
            t = Count(j)
            Count(j) = Count(j-1)
            Count(j - 1) = t
            s = Names(j)
            Names(j) = Names(j-1)
            Names(j - 1) = s
        END IF
    NEXT j
NEXT i
REM определение порога для количества появлений
REM задач из списка вывода; порог равен Count(j)
IF Num >= 3 THEN
    j = 3

```

```

ELSE
  j = Num
END IF
i = 1
REM Вывод наиболее популярных задач
WHILE i <= Num AND Count(i) >= Count(j)
  PRINT Names(i), Count(i)
  i = i + 1
WEND

```

Задачи для самостоятельного решения (OC)

C4.1

Популярная газета объявила конкурс на определение советского фильма, для которого, по мнению читателей, нужно снять продолжение. На выбор был предложен список из 10 фильмов. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет обрабатывать результаты sms-голосования по данному вопросу. Результаты голосования получены в виде списка фильмов (каждый элемент списка соответствует одному sms-сообщению). Следует учитывать, что количество голосов в списке может быть очень велико. Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подаётся количество пришедших sms-сообщений N. В каждой из последующих N строк записано название фильма в виде текстовой строки. Длина строки не превосходит 100 символов, название фильма может содержать буквы, цифры, пробелы и знаки препинания.

Пример входных данных:

```

6
Белое солнце пустыни
Бриллиантовая рука
Белое солнце пустыни
Белое солнце пустыни
Гараж
Бриллиантовая рука

```

Программа должна вывести список всех фильмов, встречающихся в списке, в порядке убывания (невозрастания) количества голосов, отданных за этот фильм, с указанием количества отданных за него голосов. При этом название каждого фильма должно быть выведено ровно один раз, вне зависимости от того, сколько голосов было отдано за этот фильм.

Пример выходных данных для приведённого выше примера входных данных:

```

Белое солнце пустыни 3
Бриллиантовая рука 2
Гараж 1

```

C4.2

В командных олимпиадах по программированию для решения предлагается не больше 12 задач. Команда может решать предложенные задачи в любом порядке. Подготовленные решения команда посылает в единую проверяющую систему соревнований. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать пришедшие запросы на проверку, чтобы определить популярность той или иной задачи. Следует учитывать, что количество запросов в списке может быть очень велико, так как многие соревнования проходят с использованием Интернет.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подается количество пришедших запросов N. В каждой из последующих N строк записан номер задачи от 1 до 12.

Пример входных данных:

6
1
2
1
1
5
2

Программа должна вывести список всех задач, встречающихся в запросах, в порядке возрастания (неубывания) количества запросов на проверку той или иной задачи с указанием количества запросов по ней. При этом каждая задача должна быть выведена ровно один раз, вне зависимости от того, сколько раз она встречается в списке.

Пример выходных данных для приведённого выше примера входных данных:

5 1
2 2
1 3

C4.3

На вход программе подаются сведения о пассажирах, сдавших свой багаж в камеру хранения. В первой строке задано текущее время: через двоеточие два целых числа, соответствующие часам (от 00 до 23 – ровно 2 символа) и минутам (от 00 до 59 – ровно 2 символа). Во второй строке сообщается количество пассажиров N , которое не меньше 10, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат:

<Фамилия> <время освобождения ячейки>,

где <Фамилия> – строка, состоящая не более, чем из 20 символов, <время освобождения ячейки> – через двоеточие два целых числа, соответствующие часам (от 00 до 23 – ровно 2 символа) и минутам (от 00 до 59 – ровно 2 символа). <Фамилия> и <время освобождения ячейки> разделены одним пробелом. Сведения отсортированы в порядке времени сдачи багажа.

Требуется написать на любом языке программирования программу, выводящую фамилии пассажиров, которые в ближайшие 2 часа должны освободить ячейки, в хронологическом порядке освобождения ячеек.

Пример входных данных:

10:00

3

Иванов 12:00

Петров 10:00

Сидоров 12:12

Результат работы программы для этого примера

Петров

Иванов

C4.4

На автозаправочных станциях (АЗС) продается бензин с маркировкой 92, 95 и 98. В городе М был проведен мониторинг цены бензина на различных АЗС.

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет определять для бензина с маркировкой 92, на какой АЗС его продают по второй по минимальности цене (считается, что самой низкой цене потребители не доверяют), а если таких АЗС несколько, то выдается только количество таких АЗС. Если все АЗС, у которых 92-й бензин есть, продают его по одной и той же цене, то эта цена считается искомой и выдается либо число таких АЗС, когда их несколько, либо конкретная АЗС, если она одна. Гарантируется, что хотя бы одна АЗС 92-й бензин продает.

На вход программе сначала подается число данных о стоимости бензина N . В каждой из следующих N строк находится информация в следующем формате:

<Компания> <Улица> <Марка> <Цена>

где <Компания> – строка, состоящая не более, чем из 20 символов без пробелов, <Улица> – строка, состоящая не более, чем из 20 символов без пробелов, <Марка> – одно из чисел – 92, 95 или 98, <Цена> – целое число в диапазоне от 1000 до 3000, обозначающее стоимость одного литра бензина в копейках. <Компания> и <Улица>, <Улица> и <Марка>, а также <Марка> и <цена> разделены ровно одним пробелом. Пример входной строки:

МигОйл Мичуринский 92 1950

Программа должна выводить через пробел Компанию и Улицу искомой АЗС или их количество, если искомым вариантов несколько. Пример выходных данных:

ТНК Можайский

Второй вариант выходных данных:

4

C4.5

По каналу связи передаётся последовательность положительных целых чисел, все числа не превышают 1000. Количество чисел известно, но может быть очень велико. Затем передаётся контрольное значение последовательности – наибольшее число R , удовлетворяющее следующим условиям:

1) R – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных элементов последовательности, равных по величине, допускаются);

2) R делится на 22.

Если такого числа R нет, то контрольное значение полагается равным 0.

В результате помех при передаче как сами числа, так и контрольное значение могут быть искажены.

Напишите эффективную, в том числе по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет проверять правильность контрольного значения. Программа должна напечатать отчёт по следующей форме:

Вычисленное контрольное значение: ...

Контроль пройден (или – Контроль не пройден)

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел N . В каждой из последующих N строк записано одно натуральное число, не превышающее 1000. В последней строке записано контрольное значение.

Пример входных данных:

6

55

997

22

7

9

400

22000

Пример выходных данных для приведённого выше примера входных данных:

Вычисленное контрольное значение: 22000

Контроль пройден

C4.6

Сейсмограф автоматической геофизической станции «Токанава» передаёт показания каждую минуту. Показания сейсмографа – положительные вещественные числа, не превышающие 1000.

Необходимо найти в заданной серии показаний прибора максимальное произведение двух показаний, между моментами передачи которых прошло не менее 7 минут. Общее количество показаний прибора в серии не превышает 10 000. Временем передачи сигнала можно пренебречь.

Напишите на любом языке программирования программу для решения поставленной задачи. Ваша оценка будет зависеть не только от правильности программы, но и от того, насколько она эффективна.

Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.

Максимальная оценка за правильную программу, неэффективную ни по времени, ни по памяти, – 2 балла.

Перед программой укажите версию языка и кратко опишите использованный алгоритм.

В первой строке задаётся число N – общее количество показаний прибора. Гарантируется, что $N > 7$. В каждой из следующих N строк задаётся одно положительное вещественное число – очередное показание прибора.

Пример входных данных:

```
12
12
5
45
46
8
9
10
11
20
30
28
26
```

Программа должна вывести одно число – описанное в условии произведение.

Пример выходных данных для приведённого выше примера входных данных:

```
1350
```

3. Информационные и коммуникационные технологии

3.1. Файловые системы

Во всех операционных системах имеющаяся на компьютере информация хранится в виде файлов. Каждый файл имеет имя и располагается на определенном устройстве хранения информации. В виде файлов хранятся программы (такие файлы называются выполняемыми) и документы. Иногда в состав одного приложения или документа входят несколько файлов. Для удобства хранения и поиска файлов они объединены в папки. Папки могут быть вложены друг в друга, образуя многоуровневую древовидную структуру.

Синонимами термина “папка”, принятого в Windows, являются слова “каталог” и “директория”. Подобно файлам папки имеют свои имена.

Полное имя файла состоит из пути к файлу и имени файла. Путь к файлу представляет собой перечень имен каталогов (папок), которые нужно последовательно посетить, чтобы спуститься к файлу с самого высокого уровня дерева файлов. Во многих операционных системах (в том числе семейства Windows) полное имя файла начинается с имени устройства внешней памяти, на котором он расположен. Имя устройства образуется двумя символами – латинской буквой и следующим за ней двоеточием. После имени устройства следуют имена папок, разделенные символом «/».

Имя файла состоит из собственно имени и необязательного расширения, которое обычно означает тип файла и ассоциируется с программой, которая файлы этого типа обрабатывает. Имя и расширение разделяются точкой.

Задачи по этой теме касаются либо построения дерева каталогов по полному имени файла, либо полного имени по описанию структуры каталогов. Также к этой теме относятся задания на отбор файлов из списка, соответствующих той или иной маске имени файла. Рассмотрим решение типовых задач по этой теме.

Пример:

Дано дерево каталогов

```
A: - [ DOC3 ] - Doc1
      | [ TOM3 ] - Doc3
```

Определите полное имя файла Doc3.

- 1) A:/DOC3
- 2) A:/DOC3/Doc3
- 3) A:/ DOC3/Doc1
- 4) A:/TOM3/Doc3

Решение:

Путь к файлу **Doc3** состоит из следующих элементов:

Имя устройства внешней памяти – **A:**

Имя каталога – **TOM3**

Таким образом, полное имя файла Doc3: **A:/TOM3/Doc3**, что соответствует пункту 4 в списке ответов.

Ответ: 4

Пример:

В некотором каталоге хранился файл `txt.doc`. После того, как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл `txt.doc`, полное имя файла стало `P:\doc\txt\letter\txt.doc`. Каково было полное имя этого файла до перемещения?

Решение:

По условию задачи файл должен находиться во вновь созданном каталоге. По полному имени файла видно, что он находится в каталоге **letter**, следовательно, **letter** и есть вновь созданный каталог. Каталог **letter** находится в каталоге с полным именем `P:\doc\txt`. По условию задачи файл изначально хранился в том каталоге, где был создан подкаталог **letter**, следовательно, полное имя файла было `P:\doc\txt\txt.doc`

Ответ: `P:\doc\txt\txt.doc`

Пример:

Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, какое из указанных имен файлов удовлетворяет маске:

`?hel*lo.c?*`.

1) `hello.c`

2) `hello.cpp`

3) `hhelolo.cpp`

4) `hhelolo.c`

Решение:

Поскольку в начале маски стоит знак «?» перед буквой **h**, то в имени буква **h** должна стоять на втором месте. Поскольку в маске после буквы **c** тоже стоит знак «?», то после этой буквы в расширении имени файла должен находиться непустой символ. Этим условиям из предложенных имен файлов одновременно удовлетворяет только третье - **hhelolo.cpp**. Нетрудно убедиться, что оно также соответствует остальным символам маски.

Ответ: 3

В ЕГЭ 2015 г. знания о файловой системе организации данных проверяются заданием 3.

3 (из демоверсии 2012 г.)

Для групповых операций с файлами используются **маски имён файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которой также могут встречаться следующие символы.

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

В каталоге находятся пять файлов:

`fort.docx`

`ford.docx`

`lord.doc`

`orsk.dat`

`port.doc`

Определите, по какой из масок из них будет отображена указанная группа файлов:

`fort.docx`

`ford.docx`

`lord.doc`

`port.doc`

- 1) *o?*d?*
- 2) ?o*?.d*
- 3) *or*.doc?
- 4) ?or?.doc?

Решение:

Это задание моделирует распространенную ситуацию, когда необходимо из всего множества файлов, находящихся в каталоге, отобрать группу, имеющих схожее по каким-то признакам, имя. В данном случае «за бортом» оказался файл **orsk.dat**. Он отличается тем, что перед буквосочетанием «**or**» в его имени нет символов, а у остальных отобранных файлов – ровно один символ. По этому признаку отбрасываются варианты ответа под номерами 1) и 3). Из оставшихся двух, ответ номер 4) не подходит потому, что предполагает ровно четыре символа в расширении имени файла, то есть третий и четвертый файлы из списка под эту маску не подходят. Остается убедиться, что все 4 файла из списка подходят под маску **?o*?.d***, а файл **orsk.dat** под неё, соответственно, не подходит.

Ответ: 2

Задачи для самостоятельного решения

3.1

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, какое из указанных имен файлов удовлетворяет маске:

?el*.t*xt

- 1) spell.txt
- 2) el.txt
- 3) pencil.text
- 4) welcome.text

3.2

В некотором каталоге хранился файл **taskbook.txt**. После того, как в этом каталоге создали новый подкаталог и переместили файл **taskbook.txt** в созданный подкаталог, полное имя файла стало **C:\docs\school\math\taskbook.txt**. Какое было полное имя данного файла до перемещения?

- 1) taskbook.txt
- 2) C:\docs\school
- 3) C:\docs\school\taskbook.txt
- 4) math\taskbook.txt

3.3 (ОС)

Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

- cobol.doc
- bottom.dot
- common.doc
- blossom.dot
- 1) *o?.do?
- 2) *o??o?.*
- 3) ?o*o?.do?
- 4) ?o?.???

3.4 (ОС)

Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

- 0999.txt
- 9909.ppt
- 0990.txt
- 1990.ppt
- 1) ?0*.*t
- 2) *99*.*t*
- 3) *9*9*.*t?
- 4) ?9*.*?

3.5 (ОС)

Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

- suprum.docx
- cinema.doc
- common.docx
- clame.doc
- 1) c?m?.doc?
- 2) *m*.????
- 3) *.d?c*
- 4) *c*m*.*c?

3.2. Обработка графической информации

Существует два способа кодирования изображений: векторный и растровый. Растровое изображение представляет собой совокупность точек (пикселей). Цвет каждого пикселя кодируется определенным образом, объем памяти, необходимый для хранения информации о цвете пикселя, определяет количество цветов в палитре. Наиболее распространенными являются палитры, в которых для хранения пикселя отводится 1 бит (монохромное, черно-белое изображение), 8 бит (256 оттенков серого цвета), 16 бит (65 тысяч цветов) и 24 бита

(16 млн. цветов). В частности, цвет веб-страницы кодируется как информация об интенсивности трех цветовых компонент каждого пикселя. Для успешного решения задач этой темы необходимо понимать взаимосвязь информационного объема растрового изображения, его пространственного и цветового разрешения, а также принципы кодирования цвета элементов раstra.

Векторное кодирование графических изображений представляет собой информацию о способах построения изображения в виде совокупности команд определенного графического редактора (то есть исполнителя алгоритмов). В заданиях ЕГЭ таким исполнителем является Чертежник.

Пример:

Для хранения растрового изображения размером 64x64 пикселя отвели 1,5 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

- 1) 8
- 2) 2
- 3) 16
- 4) 4

Решение:

Вычислим, сколько бит отводится для хранения одного пикселя, учитывая, что в одном килобайте $1024 = 2^{10}$ байта, а в одном байте $8 = 2^3$ бит. Для этого информационный объем изображения, выраженный в битах, разделим на количество пикселей:

$$1,5 \cdot 1024 \cdot 8 / (64 \cdot 64) = 1,5 \cdot 2^{10} \cdot 2^3 / (2^6 \cdot 2^6) = 1,5 \cdot 2^{10-3-6-6} = 1,5 \cdot 2 = 3.$$

Итак, на один пиксель приходится три бита. Три бита позволяют закодировать максимум $2^3 = 8$ различных значений. Поэтому максимально возможное число цветов в палитре изображения равно 8, что соответствует ответу под номером 1.

Ответ: 1.

Пример:

Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 256x256 пикселей, если известно, что в изображении используется палитра из 2^{16} цветов. Саму палитру хранить не нужно.

- 1) 128
- 2) 512
- 3) 1024
- 4) 2048

Решение:

Найдем минимальный объем памяти, необходимый для хранения одного пикселя.

В изображении используется палитра из 2^{16} цветов, следовательно, одному пикселю может быть сопоставлен любой из 2^{16} возможных номеров цвета в палитре. Поэтому минимальный объем памяти для одного пикселя будет равен $\log_2(2^{16}) = 16$ битам. Минимальный объем памяти, достаточный для хранения всего изображения будет равен $16 \cdot 256 \cdot 256 = 2^4 \cdot 2^8 \cdot 2^8 = 2^{20}$ бит = 2^{17} байт = 2^7 килобайт = 128 килобайт, что соответствует пункту под номером 1.

Ответ: 1.

Пример:

Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor="#FFFFFF">`?

- 1) белый
- 2) зеленый
- 3) красный
- 4) синий

Решение:

В 24-битной системе кодирования цветов RGB, используемой для мониторов, каждая из трех цветовых составляющих (красная, зеленая, синяя) может принимать значение от 0 до 255 и кодируется одним байтом (две шестнадцатеричные цифры). 0 соответствует полному отсутствию данной цветовой составляющей в пикселе, 255 (FF_{16}) – максимальной яркости данного цвета. Максимальная яркость всех трех цветовых сигналов $FFFFFF_{16}$ в RGB обеспечивает белый цвет пикселя. Минимальная яркость (все цвета выключены) - 000000_{16} - черный.

Ответ: 1**Пример:**

Исполнитель Чертежник умеет выполнять следующие команды

*поднять перо**опустить перо*

сместиться в точку (x,y) – переместиться с поднятым или опущенным пером из текущей точки на листе в заданную.

Изначально, текущей считается точка с координатами (0,0), совпадающая с центром прямоугольного листа бумаги. Координатные оси направлены вдоль краев листа. Масштаб единиц по обеим осям одинаковый. После выполнения команды текущей считается новая позиция пера.

Для Чертежника задан алгоритм, в котором пропущено два одинаковых элемента (вместо них – подчеркивание):

использовать Чертежник**алг** фигура**нач**

вещ A, L, S

цел N, I

A := 0

L := 20

N := 100

S := $2 * 3.14 / N$ *поднять перо**сместиться в точку*(_____)*опустить перо***нц** для I от 1 до N

A := A + S

сместиться в точку(_____)**кц****кон**

Какой элемент следует вставить в скобки вместо подчеркивания так, чтобы после выполнения алгоритма появилось приближенное изображение окружности?

1) $L + \cos(A)$, $L + \sin(A)$ 2) $L * \cos(S)$, $L * \sin(S)$ 3) $L * L$, $A * A$ 4) $L * \cos(A)$, $L * \sin(A)$ **Решение:**

По приведенному в задаче алгоритму видно, что будет построена ломаная линия из 100 отрезков. Переменная A внутри цикла каждый раз прирастает на $1/100$ от радианной меры окружности, то есть представляет собой величину угла между двумя радиусами: от центра окружности к первоначально поставленной точке и к текущей точке. Ясно, что проекции радиуса на оси будут пропорциональны косинусу и синусу этого угла. L в данном алгоритме – радиус, для определения координаты следующей точки следует радиус умножить на

косинус и синус угла соответственно. Вариант ответа 4) даст изображение ломаной линии, приближенной к окружности с радиусом в 20 точек с центром в начале координат. Если вставить в пропущенные места ответ 1), на экране появится вырожденная окружность радиусом в 1 точку с центром в точке (20,20). Вариант 2) приведет к тому, что на экране появится ровно одна точка, так как переменная S в цикле не изменяется. Вариант ответа 3) приведет к построению отрезка от точки (400, 0) до точки (400, 39). Следует обратить внимание, что речь в любом случае идет о компьютерном исполнителе, поэтому построенные фигуры будут дискретными. В данной модели дискретность определяется размерами точки, построение отрезка длиной менее одной точки невозможно. Именно поэтому ответ 1) не может быть рассмотрен как верный, фактически окружность построена не будет. Правильный ответ – 4).

Ответ: 4

Задачи для самостоятельного решения

Разд. 3.2.1

Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor="#FF00FF">`?

- 1) красный
- 2) черный
- 3) зеленый
- 4) фиолетовый

Разд. 3.2.2

Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor="#0000FF">`?

- 1) синий
- 2) зеленый
- 3) черный
- 4) красный

Разд. 3.2.3 (ОС)

Исполнитель Чертежник умеет выполнять следующие команды

поднять перо

опустить перо

сместиться в точку (x,y) – переместиться с поднятым или опущенным пером из текущей точки на листе в заданную.

Изначально, текущей считается точка с координатами (0,0) совпадающая с центром прямоугольного листа бумаги. Координатные оси направлены вдоль краев листа. Масштаб единиц по обеим осям одинаковый. После выполнения команды текущей считается новая позиция пера.

Для Чертежника задан алгоритм, в котором пропущен один элемент (вместо него – подчеркивание):

использовать Чертежник

алг фигура

нач

```
. вещ A, L, S
. цел N, I
. A := 0
. L := 20
. N := 100
. S := 3.14/N
. поднять перо
. сместиться в точку (0 , 0)
. опустить перо
. нц для I от 1 до N
. . A:=A+S
. . сместиться в точку ( _____ )
. кц
```

кон

Какой элемент следует вставить в скобки вместо подчеркивания так, чтобы после выполнения алгоритма появилось приближенное изображение фрагмента параболы?

- 1) S*S
- 2) L ,A*L*L
- 3) A*A
- 4) A ,L*A*A

3.3. Цифровое кодирование звука

Звук представляет собой колебания давления на барабанную перепонку – орган слуха человека. Амплитуда этих колебаний представляет собой громкость звука, частота колебаний – его высоту. При аналоговой записи звуковые колебания порождаются либо колебаниями иглы звукоснимателя при движении в бороздке грампластинки, либо электромагнитными колебаниями, записанными на магнитофонную ленту. При цифровой записи звуковой сигнал дискретизируется по времени. Количество мгновенных значений амплитуды сигнала за единицу времени называется частотой дискретизации сигнала. Так, например, частота дискретизации в 40 кГц означает, что в секунду фиксируется 40 000 мгновенных значений амплитуды сигнала.

Поскольку под хранение числа – величины мгновенного значения амплитуды - мы можем отвести только конечное число двоичных разрядов (и, по техническим соображениям, одинаковое для каждого значения), то значения амплитуды квантуются (нормируются и округляются до ближайших целых значений в соответствии с количеством отведенных для них разрядов). Например, если используется 8 разрядное представление звука, то возможно всего 256 возможных квантованных значений амплитуды. Эта величина (количество бит для записи одного мгновенного значения амплитуды) также называется глубиной кодирования звука.

Чем выше частота дискретизации и разрядность представления звука, тем более точно можно сохранить звучание исходного сигнала.

В случае записи многоканального (стерео- и квадрослука) указанный подход применяется к каждому каналу.

Качественная звукозапись требует высоких значений частоты дискретизации и, в силу этого, порождает огромные объемы информации. Поэтому большинство форматов цифровой звукозаписи предполагает использование алгоритмов сжатия данных. Однако, в заданиях

единого государственного экзамена вопросы сжатия полученных цифровых данных и применения каких-либо других оптимизирующих преобразований не рассматриваются.

В ЕГЭ 2015 г. задание, проверяющее умение определять объем памяти, необходимый для хранения звуковой и графической информации, может быть на позиции 9. В 2014 г. это задание находилось на позиции А8.

А8 (демоверсия 2012)

Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Запись длится 1 минуту, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 0.2 Мбайт
- 2) 2 Мбайт
- 3) 3 Мбайт
- 4) 4 Мбайт

Решение:

За одну минуту будет записано 16000×60 отсчетов (мгновенных значений амплитуды). Каждый отсчет займет 24 бита. Таким образом, общий объем файла составит:

$$16000 \times 60 \times 24 = (16 \times 8 \times 125) \times (4 \times 15) \times (8 \times 3) = 2^4 \times 2^3 \times 125 \times 2^2 \times 15 \times 2^3 \times 3 = 2^{12} \times 125 \times 45 \text{ бит.}$$

Так как 1 Мбайт = 1024 Кбайт = 2^{20} байт = 2^{23} бит, выполним преобразование:

$$(2^{12} \times 125 \times 45) : (2^{23}) = 125 \times 45 : 2^{11} = 125 \times 45 : 2048 = 5625 : 2048 \approx 2,7 \text{ Мбайт}$$

Следует заметить, что все приведенные выше расчеты легко выполняются без калькулятора.

Ответ: 3

Задачи для самостоятельного решения

А8.1

Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Запись длится 2 минуты, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 700 Кбайт
- 2) 5 Мбайт
- 3) 3 Мбайт
- 4) 43 Мбайт

А8.2

Производится одноканальная (моно) звукозапись с частотой дискретизации 22 кГц и 24-битным разрешением. Запись длится 2 минуты, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 1 Мбайт
- 2) 8 Мбайт
- 3) 15 Мбайт
- 4) 60 Мбайт

A8.3

Производится двухканальная (стерео) звукозапись с частотой дискретизации 22 кГц и 24-битным разрешением. Запись длится 2 минуты, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 1 Мбайт
- 2) 8 Мбайт
- 3) 15 Мбайт
- 4) 60 Мбайт

A8.4

Производится двухканальная (стерео) звукозапись с частотой дискретизации 11 кГц и 16-битным разрешением. Запись длится 4 минуты, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 10 Мбайт
- 2) 2,5 Мбайт
- 3) 5 Мбайт
- 4) 40 Мбайт

A8.5

Производится одноканальная (моно) звукозапись с частотой дискретизации 8 кГц и 8-битным разрешением. Запись длится 2 минуты, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 120 Кбайт
- 2) 7.5 Мбайт
- 3) 15 Мбайт
- 4) 940 Кбайт

3.4. Обработка информации в электронных таблицах

Знания и умения по теме «электронные таблицы» в ЕГЭ 2015 г. будет проверять задание 7. Содержание этой темы достаточно обширно и разнообразие заданий по данной теме весьма велико.

Работа с формулами, абсолютные и относительные ссылки

Электронная таблица состоит из ячеек, каждая из которых имеет адрес аналогично клеточкам шахматной доски или игрового поля для “морского боя”. Столбцы таблицы нумеруются буквами и буквосочетаниями латинского алфавита, а строки – числами. Таким образом, ячейка на пересечении 15-й строки и столбца D будет иметь адрес D15.

В ячейке может храниться не только значение, введенное пользователем, но и формула, по которой автоматически рассчитывается значение ячейки. Для того, чтобы электронная таблица могла производить вычисления по формулам, они должны быть записаны по определенным правилам. Отличительным признаком формулы является знак = в ее начале. В формулах могут использоваться адреса других ячеек и различные функции, в том числе математические *sin*, *cos*, *exp* (показательная функция с основанием, равным числу *e*), *abs* (модуль числа) и так далее.

В электронных таблицах принято следующее правило: обычные адреса ячеек в формулах являются относительными. Это означает, что при копировании ячейки, содержащей формулу, в ячейку, отстоящую от исходной на некоторое число столбцов и строк, адреса ячеек в формуле изменяются на такое же число столбцов и строк. Пусть, например, ячейка B2 содержит формулу =C2+1. При копировании ячейки формула изменится следующим образом:

	A	B	C	D
1	=B1+1		=D1+1	
2		=C2+1		=E2+1
3	=B3+1			
4			=D4+1	

Чтобы адрес не менялся при копировании, он должен быть абсолютным. В абсолютном адресе перед обозначениями строки и столбца ставится знак \$. Если знак \$ стоит только перед именем столбца, то при копировании будет сохраняться имя столбца, если перед номером строки – номер строки.

Пусть ячейка B2 содержит формулу =\$C\$2+\$C3+C\$4. При копировании ячейки формула изменится следующим образом:

	A	B	C	D
1	=\$C\$2+\$C2+B\$4		=\$C\$2+\$C2+D\$4	
2		=\$C\$2+\$C3+C\$4		=\$C\$2+\$C3+E\$4
3	=\$C\$2+\$C4+B\$4			
4			=\$C\$2+\$C5+D\$4	

Пример:

В ячейке A1 электронной таблицы записана формула =D1-\$D2. Какой вид приобретет формула после того, как ячейку A1 скопируют в ячейку B1?

- 1) =E1-\$E2
- 2) =E1-\$D2
- 3) =E2-\$D2
- 4) =D1-\$E2

Решение:

Адрес B1 получен из адреса A1 сдвигом на 1 вправо. Также изменятся все относительные адреса столбцов в адресах формулы, а именно, D1 преобразуется в E1. Адреса строк не изменятся, так как формула копируется в пределах одной строки (первой). Адрес \$D2 не изменится, так как здесь адрес столбца абсолютный, и формула приобретет вид =E1-\$D2.

Ответ: 2.

Пример:

Дан фрагмент электронной таблицы:

	A	B	C	D
1	5	2	4	
2	10	1	6	

В ячейку D2 введена формула =A2*B1+C1. В результате в ячейке D2 появится значение:

- 1) 6
- 2) 14
- 3) 16
- 4) 24

Решение:

Подставим значения ячеек в заданную формулу: $A2*B1+C1=10*2+4=24$.

Ответ: 4.

Пример:

В электронной таблице значение формулы =СУММ(B1:B2) равно 5. Чему равно значение ячейки B3, если значение формулы =СРЗНАЧ(B1:B3) равно 3?

Решение:

Диапазону B1:B3 принадлежат три ячейки B1, B2 и B3. Функция СРЗНАЧ вычисляет среднее арифметическое ячеек диапазона, поэтому $(B1+B2+B3):3=3$. Следовательно $B1+B2+B3=9$. Учитывая, что по условию $B1+B2=5$, получаем, что $B3=4$.

Ответ:4

A7 (демоверсия 2012)

В ячейке B4 электронной таблицы записана формула = \$C3*2. Какой вид приобретет формула, после того как ячейку B4 скопируют в ячейку B6?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) = \$C5 *4 2) = \$C5 *2 3) = \$C3 *4 4) = \$C1 *2

Решение:

Формула будет перемещена на две ячейки вниз, адреса столбцов не изменятся, так как формула копируется в пределах одного столбца (второго). Множитель 2, естественно, не изменится, не изменится и адрес столбца, по двум причинам: во-первых, он абсолютный, и во-вторых, перемещение происходит внутри одного столбца. Адрес строки увеличится на 2 и формула приобретет вид = \$C5 * 2.

Ответ: 2.

Задание A11 ЕГЭ 2011 года (ОС)

В динамической (электронной) таблице приведены ежеквартальные затраты 4-х строительных организаций (в миллионах рублей) и площади построенных ими помещений (в квадратных метрах). В какой из строительных организаций средние затраты за год на строительство одного квадратного метра наименьшие?

Название организации	I квартал		II квартал		III квартал		IV квартал		Всего за год	
	Затраты	Площадь	Затраты	Площадь	Затраты	Площадь	Затраты	Площадь	Затраты	Площадь
ДСК-2	15	1200	14	1100	16	1300	20	1700	65	5300
PCY-14	35	2000	40	2500	50	3000	45	2600	170	10100
CMY-78	40	3000	60	4000	50	3500	65	4500	215	15000
Трест № 7	60	4100	40	2900	100	7200	80	5800	280	20000

- 1) ДСК-2 2) PCY-14 3) CMY-78 4) Трест №7

Решение:

При использовании компьютера данная задача решается построением дополнительного столбца и заполнением его значением результата от деления затрат на площадь. Далее можно либо найти минимум в столбце с помощью соответствующей функции, либо отсортировать таблицу по возрастанию значения нового столбца. На бумажном экзамене, когда доступа к компьютеру нет, а использование калькуляторов запрещено, можно попробовать сделать прикидку результата. Видно, что в Тресте №7 затраты на 1000 кв. м. составляют 14 млн. руб., в CMY-78 – более 14 млн. руб. (при 14 млн. совокупные затраты были бы 210.млн., а они на 5 млн. больше), в PCY-14 затраты вообще очень высоки: более 16 млн.

руб. на 1000 кв.м.). И только в ДСК-2 эти затраты менее 13 млн. руб. за 1000 кв.м, то есть наименьшие из представленных четырех организаций.

Обратите внимание, что большая часть информации, представленной в таблице, в решении не используется.

Ответ: 1.

Задачи для самостоятельного решения

7.1

В ячейке В4 электронной таблицы записана формула = \$C3*A2. Какой вид приобретет формула, после того как ячейку В4 скопируют в ячейку D6?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) = \$C5 *C4 2) = \$C5 *A4 3) = \$E3 *C2 4) = \$E5 *C4

7.2

В ячейке С6 электронной таблицы записана формула = В\$3*\$A5. Какой вид приобретет формула, после того как ячейку С6 скопируют в ячейку D5?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) = В\$3*\$A5 2) = В\$3*\$A4 3) = С\$3*\$A4 4) = С\$2*\$A4

7.3

В электронной таблице значение формулы =СУММ(D2:D4) равно 15. Чему равно значение формулы =СРЗНАЧ (D1:D4), если значение ячейки D1 равно 5?

- 1) 5 2) 10 3) 20 4) 4

7.4

В электронной таблице значение формулы =СУММ(D2:D4) равно 15. Чему равно значение формулы =СУММ(D1:D4), если значение ячейки D1 равно 5?

- 1) 5 2) 10 3) 20 4) 4

7.5 (ОС)

В динамической (электронной) таблице приведены ежеквартальные затраты 4-х строительных организаций (в миллионах рублей) и площади построенных ими помещений (в квадратных метрах). В какой из строительных организаций средние затраты за год на строительство одного квадратного метра наименьшие?

Название организации	I квартал		II квартал		III квартал		IV квартал		Всего за год	
	Затраты	Площадь	Затраты	Площадь	Затраты	Площадь	Затраты	Площадь	Затраты	Площадь
ДСК-12	52	4800	60	5000	65	6000	53	5200	230	21000
РСТ-1	57	6100	79	8300	68	7100	86	8900	290	30400
СМУ-38	25	2300	23	2000	27	2400	30	3200	105	9900
УКС-2	85	8200	61	5800	140	14400	125	11600	411	40000

- 1) ДСК-12 2) РСТ-1 3) СМУ-38 4) УКС-2

Диаграммы и графики

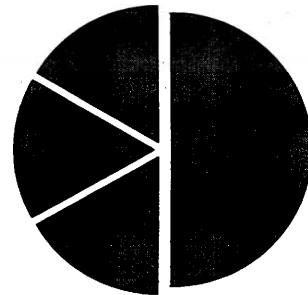
Для решения задач этой темы важно уметь правильно интерпретировать столбчатые и круговые диаграммы, учитывая, что простые столбчатые диаграммы обычно отражают абсолютные значения, а круговые – процентный состав. При интерпретации круговых диаграмм необходимо оценить соотношение величин, по которым она построена.

7 (из демоверсии 2012 г.)

Дан фрагмент электронной таблицы:

	A	B	C	D
1	3		3	2
2	$= (C1+A1) / 2$	$= C1 - D1$	$= A1 - D1$	$= B1 / 2$

Какое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?



Решение:

Диаграмма построена по четырем значениям диапазона ячеек A2:D2. Видно, что эти значения соотносятся как 1:1:1:3. При этом на круговой диаграмме неясно, какая доля круга какой ячейке диапазона соответствует. Поскольку значения ячеек A1, C1, D1 известны, заполним диапазон A2:D2 значениями вместо формул:

	A	B	C	D
1	3		3	2
2	3	1	1	$= B1 / 2$

Осталось незаполненной ячейка D2. Из приведенной в задании диаграммы ясно, что ее значение равно 1. Находим неизвестное значение ячейки B1: $B1 = 1 \cdot 2 = 2$.

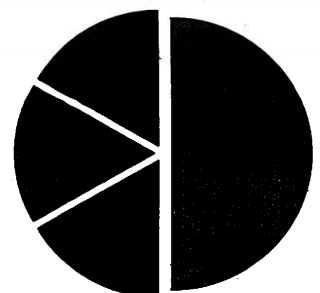
Ответ: 2.

Пример:

Дан фрагмент электронной таблицы:

	A	B	C	D
1	50		200	100
2	$= D1 / 2$	$= B1 - A1$	$= D1 - A1$	$= A1 + C1 - B1$

Какое целое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?



Решение:

Диаграмма построена по четырем значениям диапазона ячеек A2:D2. Видно, что эти значения соотносятся как 1:1:1:3. При этом на круговой диаграмме неясно, какая доля круга какой ячейке диапазона соответствует. Поскольку значения ячеек A1, C1, D1 известны, заполним диапазон A2:D2 значениями вместо формул:

	A	B	C	D
1	50		200	100
2	50	=B1-50	50	=250-B1

Остались незаполненными ячейки B2 и D2. Из приведенной в задании диаграммы ясно, что значение одной из них равно 50, а другой $50 \cdot 3 = 150$. Находим неизвестное значение ячейки B1: $B1 = 250 - 150 = 50 + 50 = 100$.

Ответ: 100.

Задачи для самостоятельного решения

7.6

Дан фрагмент электронной таблицы:

	A	B	C
1	2	1	
2	=C1-B1*2	=(B1+C1)/A1	=A1+B1

Какое целое число должно быть записано в ячейке C1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:C2 соответствовала рисунку?



7.7

Дан фрагмент электронной таблицы:

	A	B	C
1	140	70	
2	=C1-A1	=A1*2-C1	=A1-B1

Какое целое число должно быть записано в ячейке C1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:C2 соответствовала рисунку?

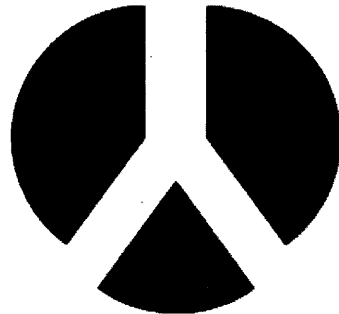


7.8

Дан фрагмент электронной таблицы:

	A	B	C
1		$= (A1 - B2) / 2$	3
2	$= C1 * 2 - B1$	2	$= C1 - B2$

Какое целое число должно быть записано в ячейке A1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:C2 соответствовала рисунку?



7.9

Дан фрагмент электронной таблицы:

	A	B	C
1		$= A1 / 2$	90
2	$= C1 - B1$	30	$= A1$

Какое целое число должно быть записано в ячейке A1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:C2 соответствовала рисунку?

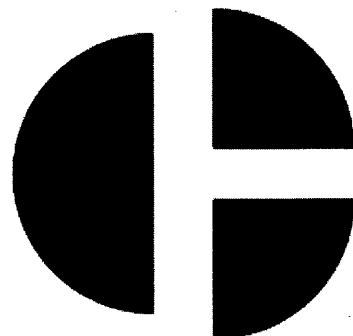


7.10

Дан фрагмент электронной таблицы:

	A	B	C
1	2	$= A1 * 3$	
2	$= (B1 - C1) * 2$	$= (B1 - A1) / 2$	$= B1 - A1$

Какое целое число должно быть записано в ячейке C1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:C2 соответствовала рисунку?



3.5. Базы данных

В 2015 г. задание 3 проверяет как знания о файловой системе организации данных, так и знания о технологии хранения, поиска и сортировки информации в базах данных. Разнообразие заданий по этой теме, использовавшихся в КИМ ЕГЭ разных лет, велико, поэтому следует ознакомиться с различными типами таких заданий. В ЕГЭ 2014 г. задание стояло на позиции А6.

Пример:

Для каждого файла в таблицу записывался исходный размер файла (поле РАЗМЕР), а также размеры архивов, полученных после применения к файлу различных архиваторов: программы WinZIP (поле ZIP), программы WinRAR (поле RAR) и программы StuffIt (поле SIT). Вот начало этой таблицы (все размеры в таблице – в килобайтах):

Имя Файла	Размер	ZIP	RAR	SIT
Аквариум.mw2	296	124	88	92
Муар.mw2	932	24	20	28

Нужно отобразить файлы, исходный размер которых больше 1 мегабайта и размер которых при использовании WinRAR уменьшился более чем в 4 раза. Для этого достаточно найти в таблице записи, удовлетворяющие условию:

- 1) (РАЗМЕР > 1000) ИЛИ (РАЗМЕР / RAR > 4)
- 2) (РАЗМЕР > 1024) И (RAR < 256)
- 3) (РАЗМЕР > 1024) И (РАЗМЕР / RAR > 4)
- 4) (РАЗМЕР > 1024) ИЛИ (РАЗМЕР / RAR > 4)

Решение:

В задачах этого типа конкретные значения полей базы данных не важны, таблица приводится только для визуализации описанной структуры базы данных. Задача состоит в том, чтобы сформировать правильный запрос по описанному условию. Ключевым здесь является условие конъюнкции (логического умножения) требований, а также соотношение 1 Мбайт = 1024 Кбайт.

Ответ: 3.

Пример:

Результаты тестирования представлены в таблице:

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Агеева	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григоренко	м	54	74	68	75	83
Доронина	ж	71	63	56	82	79
Ефимова	ж	33	25	74	38	46
Жеглова	ж	18	92	83	28	61

Сколько записей в ней удовлетворяют условию «Пол = 'ж' ИЛИ Химия > Биология»?

- 1) 5
- 2) 2
- 3) 3
- 4) 4

Решение:

В задачах этого типа конкретные значения полей базы данных играют ключевую роль. Фактически при ее решении следует исполнить соответствующий запрос и отобразить из представленных полей те, которые соответствуют условию.

При решении конкретной задачи надо обратить внимание, что два условия в запросе связаны операцией «ИЛИ». Это значит, что будут отобраны **все** записи, значением поля «Пол» которых является 'ж', а также те записи, где балл по химии больше балла по биологии. Записей об экзаменуемых женского пола в таблице четыре, а из оставшихся двух только у Воронина балл по химии (45) больше балла по биологии (23). Итак, условию удовлетворяют 5 записей из 6, что соответствует ответу под цифрой 1.

Ответ: 1.

Пример:

Результаты тестирования представлены в таблице:

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Агеева	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григоренко	м	54	74	68	75	83
Доронина	ж	71	63	56	82	79
Ефимова	ж	33	25	74	38	46
Жеглова	ж	18	92	83	28	61

Сколько записей в ней удовлетворяют условию «Пол = 'ж' И Химия > Биология»?

- 1) 5 2) 2 3) 3 4) 4

Решение:

Таблица та же самая, что и в прежней задаче, но два условия в запросе связаны операцией «И». Это значит, что будут отобраны **только те** записи, значением поля «Пол» которых является 'ж', и где балл по химии больше балла по биологии. Записей об экзаменуемых женского пола в таблице четыре, из них балл по химии больше балла по биологии только у двух: Ефимовой и Жегловой. Итак, условию удовлетворяют 2 записи из 6, что соответствует ответу под цифрой 2.

Ответ: 2.

Пример (демоверсия 2011 г.):

База данных о торговых операциях дистрибутора состоит из трех связанных таблиц. Ниже даны фрагменты этих таблиц:

Таблица зарегистрированных дилеров:

Наименование организации	ID дилера	Регион	Адрес
ООО «Вектор»	D01	Башкортостан	г. Уфа, ул. Школьная, 15
АО «Луч»	D02	Татарстан	г. Казань, ул. Прямая, 17
АОЗТ «Прямая»	D03	Адыгея	г. Майкоп, просп. Мира, 8
ООО «Окружность»	D04	Дагестан	г. Дербент, ул. Замковая, 6
ИЧП Скаляр	D05	Дагестан	г. Махачкала, ул. Широкая, 28
АО «Ромб»	D06	Татарстан	г. Набережные Челны, ул. Заводская, 4

Таблица отгрузки товара

Номер накладной	Отгружено дилеру	Артикул товара	Отгружено упаковок	Дата отгрузки
001	D01	01002	300	5/01/2009
002	D02	01002	100	5/01/2009
003	D06	01002	200	5/01/2009
004	D01	02002	20	5/01/2009
005	D02	02002	30	5/01/2009
006	D02	01003	20	6/01/2009

Таблица товаров

Наименование товара	Артикул	Отдел	Количество единиц в упаковке	Брутто вес упаковки
Фломастеры, пачка 24 шт.	01001	Канцтовары	24	5
Бумага А4, пачка 500 листов	01002	Канцтовары	5	10
Скрепки металлические 1000 шт.	01003	Канцтовары	48	20
Розетки трехфазные	02001	Электротовары	12	2
Лампа накаливания 60 вт	02002	Электротовары	100	8
Выключатель 2-клавишный	02003	Электротовары	48	7

Сколько пачек бумаги было отгружено в Татарстан 5 января 2009 г.?

- 1) 100
- 2) 200
- 3) 500
- 4) 1500

Решение:

Реляционные базы данных обычно состоят из нескольких таблиц, каждая из которых характеризует свойства определенного объекта. В данном случае объектами являются дилеры, товары и накладные (записи об отгрузке товара дилеру). Вопрос задания касается отгрузки, поэтому ответ надо искать в средней таблице.

Таблицы связаны между собой значениями ключевых полей. Записи в ключевых полях уникальны, они позволяют однозначно отобразить конкретную запись в таблице любой величины. В таблицах дилеров таким полем является «ID дилера», в таблице товаров – «Артикул», в таблице отгрузки – «Номер накладной». Таблица отгрузки связана с таблицей дилеров и таблицей товаров через значения ключевых полей. Нас интересует бумага в пачках, отгруженная в Татарстан 5 января. Бумага имеет артикул «01002». В Татарстане зарегистрировано два дилера: АО «Луч» (D02) и АО «Ромб» (D06). Накладных, связанных с этими дилерами, в таблице четыре, при этом датированы 5 января три из них (002, 003 и 005). Накладные 002 и 003 связаны с интересующим нас артикулом, третья накладная на отгрузку другого товара. Таким образом, в Татарстан 5 января бумага была отгружена по 2 накладным общим объемом 300 упаковок. Вопрос, однако, сформулирован о пачках бумаги, а не об упаковках. В нижней таблице находим строчку, относящуюся к бумаге, и читаем значение поля «Количество единиц в упаковке». Получаем значение 5 (пачек бумаги в коробке). Умножаем это число на общее количество упаковок (300) и получаем ответ: 1500 пачек бумаги. Этот ответ записан под цифрой 4.

Ответ: 4.

Пример (ОС):

Из правил соревнования по тяжелой атлетике: Тяжелая атлетика – это прямое соревнование, когда каждый атлет имеет три попытки в рывке и три попытки в толчке. Самый тяжелый вес поднятой штанги в каждом упражнении суммируется в общем зачете. Если спортсмен потерпел неудачу во всех трех попытках в рывке, он может продолжить соревнование в толчке, но уже не сможет занять какое-либо место по сумме 2-х упражнений. Если два спортсмена заканчивают состязание с одинаковым итоговым результатом, высшее место присуждается спортсмену с меньшим весом. Если же вес спортсменов одинаков, преимущество отдается тому, кто первым поднял победный вес. Таблица результатов соревнований по тяжелой атлетике:

Фамилия И.О.	Вес спортсмена	Взято в рывке	Рывок с попытки	Взято в толчке	Толчок с попытки
Айвазян Г.С.	77,1	147,5	3	200,0	2
Викторов М.П.	79,1	147,5	1	202,5	1
Гордезиани Б.Ш.	78,2	147,5	2	200,0	1
Михальчук М.С.	78,2	147,5	3	202,5	3
Пай С.В.	79,5	150,0	1	200,0	1
Шапсугов М.Х.	77,1	147,5	1	200,0	1

Кто победил в общем зачете (по сумме двух упражнений)?

- 1) Айвазян Г.С.
- 2) Викторов М.П.
- 3) Михальчук М.С.
- 4) Пай С.В.

Решение:

Тяжелая атлетика, пожалуй, единственный вид спорта, где турнирная таблица формируется на основании трех признаков: поднятого веса, собственного веса спортсмена и затраченных попыток. Данный тип задачи моделирует операцию сортировки базы данных по совокупности признаков. Первым признаком является поднятый вес, вторым признаком – вес спортсмена и, наконец, третьим признаком является номер попытки. Поднятый вес должен быть наибольшим, а собственный вес спортсмена и номер попытки – наименьшим.

Для решения представленной задачи создадим еще один столбец: суммарный поднятый вес

Фамилия И.О.	Вес атлета	Взято в рывке	Рывок попыт.	Взято в толчке	Толчок попыт.	Общий вес
Айвазян Г.С.	77,1	147,5	3	200,0	2	347,5
Викторов М.П.	79,1	147,5	1	202,5	1	350,0
Гордезиани Б.Ш.	78,2	147,5	2	200,0	1	347,5
Михальчук М.С.	78,2	147,5	3	202,5	3	350,0
Пай С.В.	79,5	150,0	1	200,0	1	350,0
Шапсугов М.Х.	77,1	147,5	1	200,0	1	347,5

Видно, что на первое место претендуют только три атлета из шести, поднявшие суммарно 350 килограмм. Из них минимальный собственный вес у М.С.Михальчука (78,2 кг). Он и является победителем в сумме двух упражнений. Его фамилия в списке под цифрой 3.

Обратим внимание, что третий признак – номер попытки – при сортировке не использовался. Это достаточно часто встречающаяся ситуация при работе с базами данных. Третий параметр сортировки начинает играть роль только в том случае, если в базе данных много записей с одинаковыми значениями полей, являющихся первым и вторым признаком для сортировки. В данном случае в таблице всего 6 строк, поэтому при сортировке третий признак работать, скорее всего, не будет.

Ответ: 3.

А6 (демоверсия 2012 г.)

В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных фамилию и инициалы бабушки Ивановой А.И.

Таблица 1

ID	Фамилия И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Петренко Н.Н.	Ж
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
	...	

Таблица 2

ID Родителя	ID Ребенка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

- 1) Петренко А.И. 2) Черных И.А. 3) Цейс Т.Н. 4) Петренко Н.Н.

Решение:

Приведенные две таблицы позволяют строить генеалогические деревья. Следует сначала определить ID Ивановой А.И. (23), затем во второй таблице найти ее родителей (85 и 13) и, соответственно, их родителей. У 13 это 82 и 95, родителей человека с ID 85 (Петренко И.Т.)

в таблице 2 нет. ID 82 принадлежит дедушке Ивановой А.И., его зовут Черных А.Н., а вот бабушка Цейс Т.Н. имеет ID 95 и стоит в списке ответов под цифрой 3.

Ответ: 3.

Задачи для самостоятельного решения

А6.1 (ОС)

Ниже приведены фрагменты таблиц базы данных участников конкурса исполнительского мастерства:

Страна	Участник
Германия	Силин
США	Клеменс
Россия	Холево
Грузия	Яшвили
Германия	Бергер
Украина	Численко
Германия	Феер
Россия	Каладзе
Германия	Альбрехт

Участник	Инструмент	Автор произведения
Альбрехт	флейта	Моцарт
Бергер	скрипка	Паганини
Каладзе	скрипка	Паганини
Клеменс	фортепиано	Бах
Силин	скрипка	Моцарт
Феер	флейта	Бах
Холево	скрипка	Моцарт
Численко	фортепиано	Моцарт
Яшвили	флейта	Моцарт

Представители скольких стран исполняют Моцарта?

- 1) 5 2) 2 3) 3 4) 4

А6.2 (ОС)

Из правил соревнования по тяжелой атлетике: Тяжелая атлетика – это прямое соревнование, когда каждый атлет имеет три попытки в рывке и три попытки в толчке. Самый тяжелый вес поднятой штанги в каждом упражнении суммируется в общем зачете. Если спортсмен потерпел неудачу во всех трех попытках в рывке, он может продолжить соревнование в толчке, но уже не сможет занять какое-либо место по сумме 2-х упражнений. Если два спортсмена заканчивают состязание с одинаковым итоговым результатом, высшее место присуждается спортсмену с меньшим весом. Если же вес спортсменов одинаков, преимущество отдается тому, кто первым поднял победный вес. Таблица результатов соревнований по тяжелой атлетике:

Фамилия И.О.	Вес спортсмена	Взято в рывке	Рывок с попытки	Взято в толчке	Толчок с попытки
Айвазян Г.С.	77,1	150,0	3	200,0	2
Викторов М.П.	79,1	147,5	1	202,5	1
Гордезиани Б.Ш.	78,2	150,0	2	200,0	1
Михальчук М.С.	78,2	152,5	3	202,5	2
Пай С.В.	79,5	–	–	202,5	1
Шапсугов М.Х.	77,1	150,0	3	202,5	3

Кто победил в толчке в этом соревновании?

- 1) Викторов М.П.
 2) Михальчук М.С.
 3) Пай С.В.
 4) Шапсугов М.Х.

A6.3

Результаты тестирования представлены в таблице

Фамилия	Пол	Математика	История	Физика	Химия	Биология
Андреев	м	80	72	68	66	70
Борисов	м	75	88	69	61	69
Васильева	ж	85	77	73	79	74
Дмитриев	м	77	85	81	81	80
Егорова	ж	88	75	79	85	75
Захарова	ж	72	80	66	70	70

Сколько записей в ней удовлетворяют условию

Пол=«ж» ИЛИ Физика<Биология

1)5 2)2 3)3 4)4

A6.4

В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных фамилию и инициалы сестры Цейс Н.А

Таблица 1

ID	Фамилия И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Черных Н.Н.	М
82	Петренко А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
	...	

Таблица 2

ID Родителя	ID Ребенка
82	85
96	42
95	85
82	13
95	13
85	42
82	10
95	10
...	...

1) Петренко А.И. 2) Черных И.А. 3) Цейс Т.Н. 4) Петренко И.Т.

A6.5

В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных сколько всего внуков и внучек у Цейс Н.А.

Таблица 1

ID	Фамилия И.О.	Пол
71	Иванова Т.Н.	Ж
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Черных Н.Н.	М
82	Петренко А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
	...	

Таблица 2

ID Родителя	ID Ребенка
82	85
96	42
95	85
82	13
95	13
85	42
10	95
10	71
...	...

3.6. Телекоммуникационные технологии

Адресация в сети Интернет

Web-страницы в Интернет размещаются на Web-серверах в виде текстовых файлов.

Входящие в состав страниц изображения, звуки и другие компоненты тоже размещаются на сервере в виде отдельных файлов. Файлы страниц и их компонентов имеют свои адреса (идентификаторы), называемые унифицированными указателями ресурсов (URL – Uniformed Resource Locator).

Пример URL-адреса: <http://www.mai.ru/chair806/index.htm>

URL-адрес начинается с обозначения протокола Интернета, с помощью которого осуществляется доступ к ресурсу. В данном случае указан HTTP – протокол передачи гипертекста (HyperText Transfer Protocol). После названия протокола следуют двоеточие и две наклонных черты. Далее указывается имя компьютера (www.mai.ru). После имени компьютера следует полное имя файла с ресурсом на сервере, символы "/" разделяют названия каталогов, которые нужно посетить, чтобы добраться до файла. В данном случае страница хранится на сервере в файле `index.htm` каталога `chair806`.

Пример:

Идентификатор некоторого ресурса сети Интернет имеет следующий вид:

<http://www.mail.ru/ftp.html>

Какая часть этого идентификатора указывает на протокол, используемый для передачи ресурса?

1) `www` 2) `ftp` 3) `http` 4) `html`

Решение:

Название протокола записывается в самом начале идентификатора ресурса. В данном случае это **http**.

Ответ: 3

Пример:

Доступ к файлу `net.edu`, находящемуся на сервере `ru.com` осуществляется по протоколу `ftp`. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла.

А	<code>ftp</code>
Б	<code>ru</code>
В	<code>://</code>
Г	<code>.edu</code>
Д	<code>.com</code>
Е	<code>net</code>
Ж	<code>/</code>

Решение:

Имя ресурса начинается с названия протокола, в данном случае – это `ftp` (буква А). Имя протокола должно отделяться от имени сервера двоеточием и двумя наклонными чертами (В). Имя сервера – `ru.com` кодируется буквами Б и Д. После имени сервера следует наклонная черта (Ж), отделяющая его от имени файла (Е,Г). Итак, имя ресурса `ftp://ru.com/net.edu`

Ответ: АВДЖЕГ.

Пример:

Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес.

В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

.64	3.13	3.133	20
А	Б	В	Г

Решение

IP-адрес не может содержать четырех- или пятизначное число, поэтому фрагмент Г должен быть на первом месте. Он также не может содержать трехзначное число 643 ($643 > 255$), поэтому фрагмент А – на последнем месте. Учитывая, что число 1333 тоже не может входить в его состав, получаем искомым IP-адрес 203.133.133.64 (ГБВА)

Ответ: ГБВА

12 (из демоверсии 2012 г.)

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP –адрес узла: 217.233.232.3

Маска: 255.255.252.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы. Точки писать не нужно.

A	B	C	D	E	F	G	H
0	3	217	233	232	244	252	255

Пример.

Пусть искомым IP-адрес 192.168.128.0, и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: HBAF

Решение:

В условии задачи четко указан алгоритм получения адреса сети: записать адрес узла сети и маску подсети в виде двоичных чисел и применить поразрядную конъюнкцию:

Адрес узла сети: $217.233.232.3 = 11011001.11101001.11101000.00000011$

Маска подсети: $255.255.252.0 = 11111111.11111111.11111100.00000000$

Поразрядная конъюнкция даст единицу при совпадении единиц и ноль в трех остальных случаях. Таким образом, применив её, получаем

Адрес узла сети: $217.233.232.0 = 11011001.11101001.11101000.00000000$

Можно заметить, что конъюнкция с 255 всегда дает число, конъюнкция с 0 – ноль. Поэтому для того, чтобы найти адрес сети, надо вычислить значения только одного октета, в данном случае – третьего.

Остается записать буквы вместо значений октетов: CDEA

Ответ: CDEA

Задачи для самостоятельного решения

12.1

Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес.

В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

134.72	20	1.2	35.
А	Б	В	Г

12.2

Доступ к файлу www.txt, находящемуся на сервере ftp.net, осуществляется по протоколу http. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла.

А	ftp
Б	http
В	://
Г	www
Д	.txt
Е	.net
Ж	/

12.3

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 217.233.232.133

Маска: 255.255.192.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы. Точки писать не нужно.

А	В	С	Д	Е	Ф	Г	Н
0	133	217	233	232	192	128	255

Пример.

Пусть искомый IP-адрес 192.168.128.0, и дана таблица

А	В	С	Д	Е	Ф	Г	Н
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: НВАФ

12.4

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 217.232.233.112

Маска: 255.255.240.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы. Точки писать не нужно.

A	B	C	D	E	F	G	H
0	224	217	233	232	112	240	255

Пример.

Пусть искомым IP-адрес 192.168.128.0, и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: **HBAF**

Поиск информации в сети Интернет

Для более точной формулировки запросов к поисковому используются логические связи И и ИЛИ. Связка И между двумя словами означает, что требуется найти страницы, содержащие одновременно и первое, и второе слово. Связка ИЛИ – что ищутся страницы, включающие хотя бы одно из указанных слов.

Пример

Каким условием нужно воспользоваться для поиска в сети Интернет информации о цветах, растущих на островах Тайвань или Хонсю (для обозначения логической операции «ИЛИ» используется символ |, а для операции «И» - символ &)?

- 1) цветы & (Тайвань | Хонсю)
- 2) цветы & Тайвань & Хонсю
- 3) цветы | Тайвань | Хонсю
- 4) цветы & (остров | Тайвань | Хонсю)

Решение:

По смыслу задачи, в любой найденной странице должно содержаться слово «цветы» и название одного из двух указанных островов. Этому условию удовлетворяют запросы 1 и 4. Запрос 2 требует одновременного присутствия на странице слов «Тайвань» и «Хонсю», что является избыточным требованием и неоправданно сужает поле для поиска. Запрос 3, напротив, его неоправданно расширяет. По запросу 3 будет найдено, например, много страниц о цветах, на которых не будет ни слова «Тайвань», ни слова «Хонсю».

Итак, выбираем между запросами 1 и 4. В запросе 4 наравне с «Тайвань» и «Хонсю» присутствует слово «остров», что также излишне расширяет круг поиска страницами о цветах и островах. Поэтому, правильный ответ – 1.

Ответ: 1.

Пример:

В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» - &.

А	законы & физика
Б	законы (физика & биология)
В	законы & физика & биология & химия
Г	законы физика биология

Решение:

Можно использовать два способа решения, один из которых основан на рассуждении, а второй предполагает использование графического представления операций над множествами в виде пересекающихся фигур.

Рассуждая логически, мы видим, что больше всего будет найдено страниц по запросу Г, так как при его исполнении будут найдены и страницы со словом «законы» (в том числе, например, и юридические), и страницы, со словом «физика», и страницы со словом «биология». Меньше всего будет найдено страниц по запросу В, так как в нем требуется присутствие всех четырех слов на искомой странице.

Осталось сравнить запросы А и Б. По запросу Б будут найдены все страницы, соответствующие запросу А, (так как в последних обязательно присутствует слово “законы”), а также страницы, содержащие одновременно слова “ физика ” и “ биология ”. Следовательно, по запросу Б будет найдено больше страниц, чем по запросу А.

Итак, упорядочив запросы по возрастанию страниц, получаем ответ: ВАБГ.

Для решения вторым способом рассмотрим множества страниц, содержащие каждое из искомых слов. Запросу X&Y будет соответствовать пересечение множеств X и Y, а запросу X | Y – их объединение. Воспользуйтесь графическим представлением действий над множествами и сравните площади получившихся фигур.

Ответ : ВАБГ

17 (из демоверсии 2012 г.)

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
Шахматы Теннис	7770
Теннис	5500
Шахматы & Теннис	1000

Какое количество страниц (в тысячах) будет найдено по запросу *Шахматы*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Решение:

Выше было сказано, что запросу X&Y соответствует пересечение множеств X и Y, а запросу X | Y – их объединение. Если по запросу *Шахматы | Теннис* было найдено 7770 страниц, то среди них были страницы, содержавшие либо оба этих слова, либо только одно из них. Так как страниц, содержащих оба эти слова, было найдено ровно 1000, то из 5500 страниц, содержащих слово «Теннис», 1000 содержит также слово «Шахматы», а 4500 – не содержат этого слова. Поэтому из общего количества 7770 страниц, надо вычесть 4500, на которых есть слово «Теннис», но нет слова «Шахматы». Полученное число в 3270 страниц и будет результатом запроса «Шахматы» и, соответственно, ответом на задание.

Ответ: 3270

Задачи для самостоятельного решения

17.1

В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» – &.

1	барокко классицизм
2	барокко (классицизм & модерн)
3	(барокко & ампир) (классицизм & модерн)
4	барокко ампир классицизм модерн

17.2

В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» – &.

1	живопись & литература
2	живопись литература
3	живопись литература графика
4	живопись & литература & графика

17.3 (OC)

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» - символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
Шахматы Теннис	7150
Шахматы	4000
Теннис	3820

Какое количество страниц (в тысячах) будет найдено по запросу
Шахматы & Теннис?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

17.4 (ОС)

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» - символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
<i>Юпитер</i>	2350
<i>Юпитер Сатурн</i>	5400
<i>Юпитер & Сатурн</i>	1100

Какое количество страниц (в тысячах) будет найдено по запросу *Сатурн*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

17.5 (Демоверсия 2015 г.)

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в сотнях тысяч)
<i>Ухо</i>	35
<i>Подкова</i>	25
<i>Наковальня</i>	40
<i>Ухо Подкова Наковальня</i>	70
<i>Ухо & Наковальня</i>	10
<i>Ухо & Подкова</i>	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу *Подкова & Наковальня*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

ОТВЕТЫ И РЕШЕНИЯ

К разделу 1. Математические основы информатики

B1.1 55; **B1.2** 240; **B1.3** 50; **B1.4** 25; **B1.5** 40;
10.1 УУУОА; **10.2** ЯЯЯУЯ; **10.3** АУАУА; **10.4** ОАААО; **10.5** УОАОУ; **10.6** 256; **10.7** 125;
10.8 1024; **10.9** РКРР
13.1 2; **13.2** 4; **13.3** 3; **13.4** 4; **13.5** 2; **13.6** 3; **13.7** 400;
1.1 2; **1.2** 3; **1.3** 1; **1.4** 4; **1.5** 4; **1.6** 4;
9.1 1040; **9.2** 648; **9.3** 3586; **9.4** 416; **9.5** 2072; **9.6** А33; **9.7** Б4;
4.1 1; **4.2** 2; **4.3** 3; **4.4** 4; **4.5** 4; **4.6** 3; **4.7** 9;
16.1 4; **16.2** 5, 6, 10, 15, 30; **16.3** 5; **16.4** 6, 14, 30; **16.5** 5, 25; **16.6** 4; **16.7** 2013;
2.1 1; **2.2** 2; **2.3** 4; **2.4** 2; **2.5** 3; **2.6** 1; **2.7** 4;
18.1 4; **18.2** 1; **18.3** 2; **18.4** 1; **18.5** 1; **18.6** 2; **18.7** 2; **18.8** 17;
23.1 2; **23.2** 8; **23.3** 64;
5.1 3; **5.2** 1; **5.3** 1; **5.4** 2; **5.5** 2;
15.1 5; **15.2** 4; **15.3** 23; **15.4** 24.

К разделу 2. Алгоритмизация и программирование

6.1 1; **6.2** 2; **6.3** 1; **6.4** 3; **6.5** 1; **6.6** 1; **6.7** 3; **6.8** 2;
6.9 121211; **6.10** 2212; **6.11** 21221; **6.12** 2212; **6.13** 1221;
22.1 8; **22.2** 4; **22.3** 9; **22.4** 5; **22.5** 37; **22.6** 25; **22.7** 96;
22.8 12; **22.9** 14; **22.10** 19; **22.11** 10; **22.12** 94; **22.13** 26; **22.14** 0;

26.1 – Решение:

Выигрывает Ваня.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделённые запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучах соответственно.

	1 ход	2 ход	3 ход	4 ход
Стартовая позиция	Петя (все варианты хода)	Ваня (выигрышные ходы)	Петя (все варианты хода, кроме непосредственно проигрышных)	Ваня (выигрышные ходы)
2, 3	2, 9	<u>2, 13</u>	2, 17	<u>2, 51</u>
			6, 13	<u>6, 39</u>
	2, 7	<u>2, 21</u>	6, 21	<u>6, 63</u>
			2, 25	<u>2, 75</u>
	6, 3	<u>18, 3</u>	18, 9	<u>54, 9</u>
			22, 3	<u>66, 3</u>
18, 7			<u>54, 7</u>	

Таблица содержит все возможные варианты ходов первого игрока. Из неё видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.

26.2 Решение:

Выигрывает Петя, своим первым ходом он должен удвоить количество камней в первой куче. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделённые запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучах соответственно.

	1 ход	2 ход	3 ход	4 ход
Позиция после первого хода	Ваня (все варианты хода)	Петя (выигрышные ходы)	Ваня (все варианты хода кроме непосредственно проигрышных)	Петя (выигрышные ходы, эквивалентному достаточно указать один из вариантов)
<u>6, 2</u>	6, 4	<u>9, 4</u>	9, 8	<u>18, 8</u> <u>9, 16</u>
			12, 4	<u>15, 4</u>
			9, 7	<u>18, 7</u>
			9, 10	<u>18, 10</u>
	6, 5	<u>9, 5</u>	12, 5	<u>15, 5</u>
			9, 8	<u>18, 8</u> <u>9, 16</u>
			9, 2	<u>18, 2</u>
	12, 2	<u>15, 2</u>	<u>Выигрыш Пети</u>	

Таблица содержит **все возможные** варианты ходов второго игрока. Из неё видно, что при любом ответе второго игрока у первого имеется ход, приводящий к победе.

26.3 – Решение:

1. а) Петя может выиграть, если $S = 13, \dots, 38$. Во всех этих случаях достаточно утроить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 38 камней.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 12$ камней. Тогда после первого хода Пети в куче будет 13 или 36 камней. В обоих случаях Ваня утраивает количество камней и выигрывает в один ход.

2. Возможные значения S : 4 и 11. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 12 камней: в первом случае утроением, во втором добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.

3. Возможное значение S : 10. После первого хода Пети в куче будет 11 или 30 камней. Если в куче станет 30 камней, Ваня утроит количество камней и выигрывает первым ходом. Ситуация, когда в куче 11 камней, уже разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчеркнуты.

Положения после очередных ходов				
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
10	10+1=11	11+1=12	12+1=13	13*3=39
			12*3=36	36*3=108
	10*3=30	30*3=90		

26.4 – Решение:

1. а) Петя может выиграть, если $S = 22, \dots 43$. Во всех этих случаях достаточно удвоить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 43 камней.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 21$ камень. Тогда после первого хода Пети в куче будет 22, 23 или 42 камня. Во всех случаях Ваня удваивает количество камней и выигрывает первым ходом.

2. Возможные значения S : 19, 20. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 21 камня: в первом случае добавлением двух камней, во втором добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выиграет.

3. Возможное значение S : 18. После первого хода Пети в куче будет 19, 20 или 38 камней. Если в куче станет 38 камней, Ваня удвоит количество камней и выиграет первым ходом. Ситуация, когда в куче 19 или 20 камней, уже разобрана в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты.

Положения после очередных ходов				
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
18	18+1=19	19+2=21	21+1=22	<u>22*2=44</u>
			21+2=23	<u>23*2=46</u>
			21*2=42	<u>42*2=84</u>
	18+2=20	20+1=21	21+1=22	<u>22*2=44</u>
			21+2=23	<u>23*2=46</u>
			21*2=42	<u>42*2=84</u>
	18*2=36	36*2=72		

26.5 – Решение:

1. а) Петя может выиграть, если $S = 16, \dots 30$. Во всех этих случаях достаточно удвоить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 30 камней.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 15$ камней. Тогда после первого хода Пети в куче будет 16, 17 или 30 камней. Во всех случаях Ваня удваивает количество камней и выигрывает первым ходом.

2. Возможные значения S : 13, 14. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 15 камней: в первом случае добавлением двух камней, во втором добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выиграет.

3. Возможное значение S : 12. После первого хода Пети в куче будет 13, 14 камней или 24 камня. Если в куче станет 24 камня, Ваня удвоит количество камней и выиграет первым ходом. Ситуация, когда в куче 13 или 14 камней, уже разобрана в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов				
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
12	12+1=13	13+2=15	15+1=16	<u>14*2=32</u>
			15+2=17	<u>15*2=34</u>
			15*2=30	<u>26*2=60</u>
	12+2=14	14+1=15	15+1=16	<u>14*2=32</u>
			15+2=17	<u>15*2=34</u>
			15*2=30	<u>26*2=60</u>
	12*2=24	<u>24*2=48</u>		

14.1 3; 14.2 4; 14.3 3; 14.4 4; 14.5 4; 14.6 2; 14.7 3;

11.1 43; 11.2 38; 11.3 59; 11.4 13; 11.5 45; 11.6 20;

B2.1 -80; B2.2 -20; B2.3 -60; B2.4 110;

8.1 2048; 8.2 9; 8.3 53; 8.4 13; 8.5 3; 8.6 5100;

19.1 3; 19.2 3; 19.3 4; 19.4 4; 19.5 4;

20.1 66; 20.2 16; 20.3 996; 20.4 555; 20.5 1000; 20.6 9990; 20.7 124;

21.1 3; 21.2 4; 21.3 2; 21.4 5; 21.5 1; 21.6 600;

C1.1.

Область	Условие 1 $x*x+y*y \leq 36$	Условие 2 $x*x+y*y \geq 9$	Программа выведет	Область обрабатывается верно
A	да	нет	не принадлежит	нет
B	да	нет	не принадлежит	да
C	да	да	принадлежит	да
D	нет	-	-	нет

Возможная доработка программы

```
if (x*x+y*y <= 1) or ((x*x+y*y <= 36) and (x*x+y*y >=9 )) then
    write('принадлежит')
else
    write('не принадлежит')
```

C1.2

Область	Условие 1 $x*x+y*y \leq 1$	Условие 2 $x*x+y*y \geq 9$	Условие 3 $x*x+y*y \leq 36$	Программа выведет	Область обрабатывается верно
A	да	нет	-	-	нет
B	нет	-	-	-	нет
C	нет	-	-	-	нет
D	нет	-	-	-	нет

Возможная доработка программы

```
if (x*x+y*y <= 1) or ((x*x+y*y <= 36) and (x*x+y*y >=9 )) then
    write('принадлежит')
else
    write('не принадлежит')
```

C1.3

Область	Условие 1 $x*x+y*y \geq 36$	Условие 2 $x*x+y*y \geq 1$	Программа выведет	Область обрабатывается верно
A	нет	-	-	нет
B	нет	-	-	нет
C	нет	-	-	нет
D	да	да	Принадлежит	да

Возможная доработка программы

```
if (x*x+y*y >= 36) or ((x*x+y*y <= 9) and (x*x+y*y >=1 )) then
    write('принадлежит')
else
    write('не принадлежит')
```

C1.4

Область	Условие 1 $x*x+y*y \leq 9$	Условие 2 $x*x+y*y \geq 1$	Программа выведет	Область обрабатывается верно
A	да	нет	не принадлежит	да
B	да	да	принадлежит	да
C	нет	-	-	нет
D	нет	-	-	нет

Возможная доработка программы

```
if (x*x+y*y >= 36) or ((x*x+y*y <= 9) and (x*x+y*y >=1 )) then
    write('принадлежит')
else
    write('не принадлежит')
```

C1.5

1. Пример: $x = 2, y = 0$. Для данной точки программа не выведет никакого сообщения.

В качестве ответа на первый вопрос подходит любая точка, у которой $x > 1$ или $y < 1 - x$, для этих точек программа не выведет никакого сообщения. Или точки, у которых $y < x^2, x > 0, x \leq 1, y > 1 - x$, для этих точек программа выведет сообщение «не принадлежит», а должна вывести сообщение «принадлежит».

2. Возможная доработка (Паскаль):

```
if (x<=1) and (y<=1-x) and ((y>=x*x) or (x>=0)) then
  write('принадлежит')
else
  write('не принадлежит')
```

Возможны и другие способы доработки.

C1.6

1. Пример: $x = 0, y = -1$. Для данной точки программа не выведет никакого сообщения.

В качестве ответа на первый вопрос подходит любая точка, у которой $y < 0$ или $x < -1$, для этих точек программа не выведет никакого сообщения. Или точки, у которых $y < x^2, y \geq 0, x < 0, x \geq -1$ для этих точек программа выведет сообщение «не принадлежит», а должна вывести сообщение «принадлежит».

2. Возможная доработка (Паскаль):

```
if (y>=0) and (x>=-1) and ((y>=x*x) or (x<=0)) then
  write('принадлежит')
else
  write('не принадлежит')
```

Возможны и другие способы доработки.

C1.7

1. Пример: $x = -1, y = 0$. Для данной точки программа не выведет никакого сообщения.

В качестве ответа на первый вопрос подходит любая точка, у которой $y < -x$ или $y > 1$, для этих точек программа не выведет никакого сообщения. Или точки, у которых $x^2 + y^2 > 1, y \leq 1, y \geq -x, x < 0$, для этих точек программа выведет сообщение «не принадлежит», а должна вывести сообщение «принадлежит».

2. Возможная доработка (Паскаль):

```
if (y>=-x) and (y<=1) and ((x*x+y*y<=1) or (x<=0)) then
  write('принадлежит')
else
  write('не принадлежит')
```

Возможны и другие способы доработки.

C1.8

Решение использует запись программы на Паскале.

1. Программа выведет два числа: 2 и 40.

2. Пример последовательности, содержащей чётные числа, для которой программа работает правильно: 1 3 5 20.

Объяснение. В конце работы программы значение переменной *sum* всегда равно последнему чётному числу или 0, если в последовательности нет чётных чисел. Значение переменной *count* вычисляется правильно, учитывая вторую ошибку (см. ниже), программа будет работать верно, если в последовательности сумма чётных чисел равна последнему чётному числу и это число не равно нулю.

3. В программе есть две ошибки.

Первая ошибка: неверное присваивание при вычислении текущей суммы.

Строка с ошибкой:

```
sum := x
```

Верное исправление:

```
sum := sum + x
```

Вторая ошибка: неверная проверка наличия чётных чисел.

Строка с ошибкой:

```
if sum > 0 then
```

Верное исправление:

```
if count > 0 then
```

C1.8

Решение использует запись программы на Паскале.

1. Программа выведет два числа: 2 и 4.

2. Пример последовательности, содержащей чётные числа, для которой программа работает правильно: 20 21 23 25.

Объяснение. В конце работы программы значение переменной *maxitit* всегда равно минимальному чётному числу или 1000, если в последовательности нет чётных чисел. Соответственно, программа будет работать верно, если в последовательности минимальное чётное число равно максимальному. Выведенное количество чётных чисел будет правильным в любом случае.

3. В программе есть две ошибки.

Первая ошибка: неверная инициализация *maximum*.

Строка с ошибкой:

```
maximum := 1000;
```

Верное исправление:

```
maximum := 0;
```

Вместо 0 может быть использовано любое отрицательное число.

Вторая ошибка: неверное условие при вычислении максимума.

Строка с ошибкой:

```
if x < maximum then
```

Верное исправление:

```
if x > maximum then
```

Возможно также исправление:

```
if x >= maximum then
```

C2.1

На языке Паскаль

```
max:=5;
for i:=1 to N do
    if (a[i]>0) and (a[i] mod 5=0) and (a[i]>max) then
        max:=a[i];
writeln(max);
```

На языке Бейсик

```
MAX = 5
FOR I = 1 TO N
IF A(I)>0 AND A(I) MOD 5=0 AND A(I)>MAX THEN
MAX = A(I)
ENDIF
NEXT I
PRINT MAX
```

На языке СИ

```
max=5;
for (i=0; i<N; i++)
    if (a[i]>0 && a[i]%5==0 && a[i]>max)
        max=a[i];
printf("%d", max);
```

На естественном языке

Записываем в переменную MAX начальное значение, равное 5. В цикле от первого элемента до двадцатого находим остаток от деления элемента исходного массива на пять. Если остаток от деления на пять равен нулю и значение текущего элемента массива положительно, то сравниваем значение текущего элемента массива со значением переменной MAX. Если текущий элемент массива больше MAX, то записываем в MAX значение этого элемента массива. Переходим к следующему элементу.
После завершения цикла выводим значение переменной MAX.

C2.2

На языке Паскаль

```
min:=998;
for i:=1 to N do
    if (a[i] mod 2=0) and (a[i] mod 5<>0) and (a[i]<min) then
        min:=a[i];
writeln(min);
```

На языке Бейсик

```
MIN = 998
FOR I = 1 TO N
IF A(I) MOD 2=0 AND A(I) MOD 5<>0 AND A(I)<MIN THEN
MIN = A(I)
ENDIF
NEXT I
PRINT MIN
```

На языке СИ

```
min=998;
for (i=0; i<N; i++)
    if (a[i]%2==0 && a[i]%5!=0 && a[i]<min)
        min=a[i];
printf("%d", min);
```

На естественном языке

Записываем в переменную MIN начальное значение, равное 998. В цикле от первого элемента до двадцатого находим остаток от деления элемента исходного массива на два и на пять. Если остаток от деления на два равен нулю и остаток от деления на пять не равен нулю, то сравниваем значение текущего элемента массива со значением переменной MIN. Если текущий элемент массива меньше MIN, то записываем в MIN значение этого элемента массива. Переходим к следующему элементу.

После завершения цикла выводим значение переменной MIN.

C2.3

На языке Паскаль

```
min:=10;
for i:=1 to N do
  if (a[i]>=4) and (a[i]<min) then
    min:=a[i];
writeln(min);
```

На языке Бейсик

```
MIN = 10
FOR I = 1 TO N
IF A(I) >= 4 AND A(I) < MIN THEN
MIN = A(I)
ENDIF
NEXT I
PRINT MIN
```

На языке СИ

```
min=10;
for (i=0; i<N; i++)
  if (a[i]>=4 && a[i]<min)
    min=a[i];
printf("% d", min);
```

На естественном языке

Записываем в переменную MIN начальное значение, равное 10. В цикле от первого элемента до двадцатого сравниваем элементы исходного массива с 4. Если текущий элемент больше или равен 4, то сравниваем значение текущего элемента массива со значением переменной MIN. Если текущий элемент массива меньше MIN, то записываем в MIN значение этого элемента массива. Переходим к следующему элементу.

После завершения цикла выводим значение переменной MIN.

C2.4

На языке Паскаль

```
min:=100;
for i:=1 to N do
  if (a[i]>=20) and (a[i]<min) then
    min:=a[i];
writeln(min);
```

На языке Бейсик

```
MIN = 100
FOR I = 1 TO N
IF A(I) >= 20 AND A(I) < MIN THEN
MIN = A(I)
ENDIF
NEXT I
PRINT MIN
```

На языке СИ

```
min=100;
for (i=0; i<N; i++)
  if (a[i]>=20 && a[i]<min)
    min=a[i];
printf("% d", min);
```

На естественном языке

Записываем в переменную MIN начальное значение, равное 100. В цикле от первого элемента до тридцатого сравниваем элементы исходного массива с 20. Если текущий элемент больше или равен 20, то сравниваем значение текущего элемента массива со значением переменной MIN. Если текущий элемент массива меньше MIN, то записываем в MIN значение этого элемента массива. Переходим к следующему элементу.

После завершения цикла выводим значение переменной MIN.

C2.5

На языке Паскаль

```
k := 0;
for i := 1 to N - 1 do
  if ((a[i]+a[i+1]) mod 2<>0) and (a[i]+a[i+1]<0) then
    inc(k);
writeln(k);
```

На алгоритмическом языке

```
к := 0
нц для i от 1 до N-1
  если mod(a[i]+a[i+1],2)<>0 и a[i]+a[i+1]<0
  то
    к := к + 1
  все
кц
вывод к
```

На языке Бейсик

```
K = 0
FOR I = 1 TO N - 1
  IF (A(I)+A(I+1)) MOD 2<>0 AND A(I)+A(I+1)<0 THEN
    K = K + 1
  END IF
NEXT I
PRINT K
```

На языке Си

```
k = 0;
for (i = 0; i < N - 1; i++)
  if ((a[i]+a[i+1])%2!=0 && a[i]+a[i+1]<0)
    k++;
printf("%d", k);
```

На естественном языке

Записываем в переменную K начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления суммы текущего и следующего элементов массива на 2. Если значение данного остатка не равно 0 и сумма текущего и следующего элементов массива меньше 0, увеличиваем переменную K на единицу.

После завершения цикла выводим значение переменной K

Программа читает все входные данные один раз, не запоминая их в массиве, размер которого равен N , а составляя только список встретившихся фильмов и количества голосов, отданных за каждый из них. Во время чтения данных просматривается список ранее сохраненных фильмов; если фильм уже есть в списке, то количество голосов, отданных за него, увеличивается на 1, иначе фильм добавляется в массив встретившихся фильмов (при корректных данных он не может быть больше 10). После окончания ввода производится сортировка массивов фильмов и количества голосов, отданных за них, в порядке убывания количества голосов, затем выводится список фильмов с указанием частоты встречаемости. Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая. Ниже приведены примеры решения задания на языках Паскаль и Бейсик. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования. Так, на языке C++ при считывании строковой переменной будет считано не всё название фильма, а только его первое слово, поэтому следует использовать функцию `getline(cin, s)`, аналогичная проблема возникает и в языке Си.

Пример правильной и эффективной программы на языке Паскаль:

```

Var n, Num, i, j, t: integer;
    Count: array[1..10] of integer;
    s: string;
    Names: array[1..10] of string;
Begin
    Num:=0; {Число различных фильмов в списке голосов}
    ReadLn(N); {Считываем количество голосов}
    for i:=1 to N do
    begin
        ReadLn(S); {считали название фильма}
        {Осуществляем поиск названия в списке уже встретившихся}
        j:=1;
        while (j<=Num) and (s<>Names[j]) do j:=j+1;
        {Если фильм найден}
        if j<=Num then {Увеличиваем счетчик числа голосов, отданных за
этот фильм}
            Count[j]:=Count[j]+1
        else begin {Иначе добавляем фильм в конец списка}
            Names[j]:=s;
            Count[j]:=1;
            Num:=Num+1
        end
    end;
end;
{Сортируем массивы Names и Count в порядке убывания значений массива Count}
for i:=Num downto 2 do
for j:=2 to i do if Count[j-1]<Count[j] then
begin
    t:=Count[j]; Count[j]:=Count[j-1]; Count[j-1]:=t;
    s:=Names[j]; Names[j]:=Names[j-1]; Names[j-1]:=s;
end;
for i:=1 to Num do
    WriteLn(Names[i], ' ', Count[i]);
end.

```

Пример правильной и эффективной программы на языке Бейсик:

```
DIM N, Num, i, j, t AS INTEGER
DIM Count(10) AS INTEGER
DIM Names$(10)

Num = 0 'Число различных фильмов в списке голосов
INPUT N 'Считываем количество голосов
FOR i = 1 TO N
LINE INPUT s$ 'считали название фильма
'Осуществляем поиск названия в списке уже встретившихся
j = 1
WHILE (j <= Num) AND (s$ <> Names$(j))
  j = j + 1
WEND
'Если фильм найден
IF j <= Num THEN 'Увеличиваем счетчик числа голосов, отданных за
этот фильм
  Count(j) = Count(j) + 1
ELSE ' Иначе добавляем фильм в конец списка
  Names$(j) = s$
  Count(j) = 1
  Num = Num + 1
END IF
NEXT i
'Сортируем массивы Names и Count в порядке убывания значений 'мас-
сива Count
FOR i = Num TO 2 STEP -1
FOR j = 2 TO i
IF Count(j - 1) < Count(j) THEN
  t = Count(j): Count(j) = Count(j - 1): Count(j - 1) = t
  s$ = Names$(j): Names$(j) = Names$(j - 1): Names$(j - 1) = s$
END IF
NEXT j
NEXT i
FOR i = 1 TO Num
PRINT Names$(i), Count(i)
NEXT i
END
```

C4.2

Программа читает все входные данные один раз, не запоминая все входные данные в массиве, размер которого равен N, а подсчитывая в массиве номеров задач, сколько запросов было по каждой из них. После окончания ввода производится одновременная сортировка массивов номеров задач и количества запросов, отданных за них, в порядке возрастания количества запросов, затем выводится список задач, которые хотя бы один раз встретились в списке запросов, с указанием частоты встречаемости. Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая. Ниже приведены примеры решения задания на языках Паскаль и Бейсик. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Пример правильной и эффективной программы на языке Паскаль:

```
Var n, Num, i, j, t: integer;
    Count: array[1..12] of integer;
    Names: array[1..12] of integer;
Begin
  For i := 1 to 12 do
    begin
      Count[i] := 0;
      Names[i] := i;
    end;
  ReadLn(N); {Считываем количество запросов}
  for i:=1 to N do
    begin
      ReadLn(t); {считали очередной запрос}
      Count[t] := Count[t] + 1;
    end;
  {Сортируем массивы Names и Count в порядке убывания значений
  массива Count}
  for i:=12 downto 2 do
    for j:=2 to i do if Count[j-1]>Count[j] then
      begin
        t:=Count[j]; Count[j]:=Count[j-1]; Count[j-1]:=t;
        t:=Names[j]; Names[j]:=Names[j-1]; Names[j-1]:=t;
      end;
  for i:=1 to 12 do
    if Count[i] > 0 then
      WriteLn(Names[i], ' ', Count[i]);
end.
```

Пример правильной и эффективной программы на языке Бейсик:

```
DIM N, i, j, t AS INTEGER
DIM Count(12) AS INTEGER
DIM Names(12) AS INTEGER
FOR i = 1 TO 12
  Count(i) = 0
  Names(i) = i
NEXT i
INPUT N 'Считываем количество запросов
FOR i = 1 TO N
  INPUT t 'считали задачу
  'Подсчитываем ее
  Count(t) = Count(t) + 1
NEXT i
'Сортируем массивы Names и Count в порядке убывания
'значений массива Count
FOR i = 12 TO 2 STEP -1
  FOR j = 2 TO i
  IF Count(j - 1) > Count(j) THEN
    t = Count(j): Count(j) = Count(j - 1): Count(j - 1) = t
    t = Names(j): Names(j) = Names(j - 1): Names(j - 1) = t
  END IF
```

```

NEXT j
NEXT i
FOR i = 1 TO 12
IF Count(i) > 0 THEN PRINT Names(i), Count(i)
NEXT i
END

```

C4.3

Программа верно читает входные данные, сразу запоминая в массиве только фамилии и временные окончания хранения багажа тех пассажиров, которые должны освободить ячейки в ближайшие 2 часа. Время при считывании удобно перевести в минуты и в этом же виде хранить и сравнивать. Затем полученный массив времен сортируется по неубыванию любым алгоритмом сортировки, параллельно переставляются и элементы массива с фамилиями (возможное использование одного массива записей, состоящих из двух полей). Печатаются элементы массива фамилий в полученном в результате сортировки порядке.

Пример правильной и эффективной программы на языке Паскаль:

```

type pp=record
    name:string[20];
    time:integer;
end;
var
    p:array[1..1000]of pp;
    q:pp;
    c,c1:char;
    i,j,N,time1:integer;
begin
    read(c,c1); {считаны часы текущего времени}
    time1:=60*((ord(c)-ord('0'))*10+ ord(c1)-ord('0'));
    readln(c,c,c1); {пропущено двоеточие, и считаны минуты}
    time1:=time1+(ord(c)-ord('0'))*10+ord(c1)-ord('0');
    readln(N);
    j:=1;
    for i:=1 to N do
    begin
        p[j].name:='';
        repeat
            read(c);
            p[j].name:=p[j].name+c
        until c=' '; {считана фамилия}
        read(c,c1); {считаны часы первого времени}
        p[j].time:=60*((ord(c)-ord('0'))*10+ ord(c1)-ord('0'));
        readln(c,c,c1); {пропущено двоеточие, и считаны минуты}
        p[j].time:=p[j].time+
            (ord(c)-ord('0'))*10+
            ord(c1)-ord('0');
        if (p[j].time>=time1) and
            (p[j].time<=time1+120) then
            j:=j+1; {данные занесены в массив}
        end;
    N:=j-1;
    for i:=1 to N-1 do {сортируем данные}

```

```

for j:=1 to N-i do
  if p[j].time>p[j+1].time then
    begin
      q:=p[j];
      p[j]:=p[j+1];
      p[j+1]:=q;
    end;
for i:=1 to N do
  writeln(p[i].name)
end.

```

Пример правильной программы на языке Бейсик:

```

DIM t(1000) AS INTEGER
DIM m(1000) AS STRING * 20
DIM s AS STRING
DIM nm AS STRING
LINE INPUT s
time1 = (ASC(MID$(s, 1, 1)) - ASC("0")) * 60 * 10
time1 = time1 + (ASC(MID$(s, 2, 1)) - ASC("0")) * 60
time1 = time1 + (ASC(MID$(s, 4, 1)) - ASC("0")) * 10
time1 = time1 + (ASC(MID$(s, 5, 1)) - ASC("0"))
INPUT N
k = 0
FOR j = 1 TO N
LINE INPUT s
c$ = MID$(s, 1, 1)
i = 1
WHILE NOT (c$ = " ")
  i = i + 1
  c$ = MID$(s, i, 1)
WEND
nm = MID$(s, 1, i)
time2 = (ASC(MID$(s, i + 1, 1)) - ASC("0")) * 60 * 10
time2 = time2 + (ASC(MID$(s, i + 2, 1)) - ASC("0")) * 60
time2 = time2 + (ASC(MID$(s, i + 4, 1)) - ASC("0")) * 10
time2 = time2 + (ASC(MID$(s, i + 5, 1)) - ASC("0"))
IF time2 >= time1 AND time2 <= time1 + 120 THEN
k = k + 1
t(k) = time2
m(k) = nm
ENDIF
NEXT j
FOR i = 1 TO k - 1
FOR j = 1 TO k - i
IF t(j) > t(j + 1) THEN
time2 = t(j): nm = m(j)
t(j) = t(j + 1): m(j) = m(j + 1)
t(j + 1) = time2: m(j + 1) = nm
ENDIF
NEXT j
NEXT i
FOR i = 1 TO k
PRINT m(i)
NEXT i
END

```

Программа читает все входные данные один раз, не запоминая их в массиве размер которого соответствует числу входных данных N или максимальной цене (3000). Во время чтения данных определяются две минимальных цены и количество АЗС, продающих по 92-й бензин по этим ценам. При печати результата проверяется, что у кого-то цена больше минимальной (вторая по минимальности цена существует), в этом случае искомая (искомые) АЗС – со второй по величине ценой, если это не так, то искомая (искомые) АЗС – все, продающие 92-й бензин.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая (например, когда все АЗС продают бензин по различной цене, и 92-й бензин продают не менее двух АЗС).

Ниже приведены примеры решения задания на языках Бейсик и Паскаль. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Пример правильной и эффективной программы на языке Паскаль:

```

var c: char;
    i, k, N, b, min1, min2, cnt1, cnt2: integer;
    s, s1, s2: string;
begin
  min1:=3001;
  cnt1:=0;
  readln(N);
  for i:=1 to N do
  begin
    read(c);
    s:='';
    repeat
      s:=s+c;
      read(c);
    until c=' '; {считана компания}
    repeat
      s:=s+c;
      read(c);
    until c=' '; {улица добавлена к компании}
    readln(k,b);
    if k = 92 then
    if min1 > b then
      begin
        min2:=min1; cnt2:=cnt1; s2:=s1;
        min1:=b; cnt1:=1; s1:=s
      end else
    if min1 = b then cnt1:=cnt1+1 else
    if min2 > b then
      begin
        min2:=b; cnt2:=1; s2:=s
      end else
    if min2 = b then cnt2:=cnt2+1
  end;
  if cnt2>0 then
    if cnt2=1 then writeln(s2) else writeln(cnt2)
  else {все АЗС продают 92-й бензин по одной цене}

```

```
    if cnt1=1 then writeln(s1) else writeln(cnt1);  
    writeln;  
end.
```

Пример правильной программы на языке Бейсик:

```
DIM s AS STRING  
DIM s1 AS STRING, s2 AS STRING  
min1 = 3001  
cnt1 = 0  
INPUT n  
FOR j = 1 TO n  
LINE INPUT s  
i = 0  
DO  
i = i + 1  
c$ = MID$(s, i, 1)  
LOOP WHILE c$ <> " "  
DO  
i = i + 1  
c$ = MID$(s, i, 1)  
LOOP WHILE c$ <> " "  
DO  
i = i + 1  
c$ = MID$(s, i, 1)  
LOOP WHILE c$ <> " "  
m = VAL(MID$(s, i + 1, 2))  
b = VAL(MID$(s, i + 4))  
k = i - 1  
s = LEFT$(s, k)  
IF m = 92 THEN  
IF min1 > b THEN  
min2 = min1: cnt2 = cnt1: s2 = s1  
min1 = b: cnt1 = 1: s1 = s  
ELSE  
IF min1 = b THEN  
cnt1 = cnt1 + 1  
ELSE  
IF min2 > b THEN  
min2 = b: cnt2 = 1: s2 = s  
ELSE  
IF min2 = b THEN cnt2 = cnt2 + 1  
ENDIF  
ENDIF  
ENDIF  
ENDIF  
NEXT j  
IF cnt2 > 0 THEN  
IF cnt2 = 1 THEN PRINT s2 ELSE PRINT cnt2  
ELSE  
IF cnt1 = 1 THEN PRINT s1 ELSE PRINT cnt1  
ENDIF  
END
```

Произведение двух чисел делится на 22, если:

- один из сомножителей делится на 22 (второй может быть любым) либо
- ни один из сомножителей не делится на 22, причём один из сомножителей делится на 2, а другой – на 11.

Поэтому программа, вычисляющая кодовое число, может работать так.

Программа читает все входные данные один раз, не запоминая все данные в массиве. Программа для прочитанного фрагмента входной последовательности хранит значения четырёх величин:

M2 – самое большое чётное число, не кратное 11;

M11 – самое большое число, кратное 11, но не кратное 2;

M22 – самое большое число, кратное 22;

MAX – самое большое число среди всех элементов последовательности, отличное от M22 (если число M22 встретилось более одного раза и оно же является максимальным, то MAX = M22).

*После того как все данные прочитаны, искомое контрольное значение вычисляется как максимум из произведений M22*MAX и M2*M11.*

Ниже приведён пример программы на языке Паскаль, которая реализует описанный алгоритм.

Кроме того, приведён пример программы на языке Бейсик, которая правильно решает задачу, но использует алгоритм, немного отличающийся от описанного выше. Возможны и другие правильные алгоритмы.

Допускаются решения, записанные на других языках программирования

Пример правильной и эффективной программы на языке Паскаль:

```
var M2, M11, M22, R, MAX, dat, res, i, N: longint;
begin
  M2 := 0;
  M11 := 0;
  M22 := 0;
  MAX := 0;
  readln(N);
  for i := 1 to N do
  begin
    readln(dat);
    if ((dat mod 2) = 0) and ((dat mod 11) > 0) and (dat > M2) then
      M2 := dat;
    if ((dat mod 11) = 0) and ((dat mod 2) > 0) and (dat > M11) then
      M11 := dat;
    if (dat mod 22 = 0) and (dat > M22) then
      begin
        if M22 > MAX then MAX := M22;
        M22 := dat;
      end
    else
      if dat > MAX then
        MAX := dat;
      end;
    readln(R);
    if (M2*M11 < M22*MAX) then
      res := M22*MAX
```



```

else
  res := M2*M11;
writeln('Вычисленное контрольное значение: ',res);
if R = res then writeln('Контроль пройден')
  else writeln('Контроль не пройден');
end.

```

Пример правильной и эффективной программы на языке Бейсик:

```

M22 = 0
M2 = 0
M11 = 0
MAX = 0
INPUT N
FOR I = 1 TO N
  INPUT DAT
  IF DAT MOD 2 = 0 AND DAT > M2 THEN
    M2 = DAT
  ELSE
    IF DAT MOD 11 = 0 AND DAT > M1 THEN
      M11 = DAT
    END IF
  END IF
  IF DAT MOD 22 = 0 AND DAT > M22 THEN
    IF M22 > MAX THEN
      MAX = M22
    END IF
    M22 = DAT
  ELSE
    IF DAT > MAX THEN
      MAX = DAT
    END IF
  END IF
NEXT I
INPUT R
IF M11 * M2 < M22 * MAX THEN
  RES = M22 * MAX
ELSE
  RES = M11 * M2
END IF
PRINT "Вычисленное контрольное значение:"; RES
IF RES = R THEN
  PRINT "Контроль пройден"
ELSE
  PRINT "Контроль не пройден"
END IF
END

```

Для построения программы, эффективной по времени, можно определить для каждого элемента входных данных максимальное значение от начала данных до этого элемента включительно. Затем нужно умножить каждый элемент, начиная с восьмого, на значение этого максимума, взятого на семь элементов раньше, и выбрать наибольшее из этих произведений. Предложенный алгоритм реализован в следующей программе на алгоритмическом языке

Программа 1. Пример правильной программы на алгоритмическом языке.

Программа эффективна по времени, но неэффективна по памяти

```

алг
нач
  цел s = 7 | требуемое расстояние между показаниями
  цел N
  ввод N
  вещтаб a[1:N] | все показания прибора
  цел i
  нц для i от 1 до N
    ввод a[i]
  кц
  вещтаб макс[1:N] | макс[i] - максимум из
                  | первых i показаний
  макс[1] := a[1]
  нц для i от 2 до N
    макс[i] := max(макс[i-1], a[i])
  кц
  вещ м | максимальное значение произведения
  м := 0
  нц для i от s + 1 до N
    м := max(м, a[i] * макс[i - s])
  кц
  вывод м
кон

```

Можно вместо троекратного обращения к каждому элементу делать все необходимые операции с ним сразу после чтения. В этом случае нужен только массив для хранения максимумов, а сами элементы можно не хранить. Этот путь реализован в следующей программе

Программа 2. Пример правильной программы на алгоритмическом языке.

Программа эффективна по времени, но неэффективна по памяти

```

алг
нач
  цел s = 7 | требуемое расстояние между показаниями
  цел N
  ввод N
  вещ a | очередное показание прибора
  вещтаб макс[1:N] | макс[i] - максимум из
                  | первых i показаний
  цел i
  | ввод первых показаний, допустимых пар пока нет
  ввод макс[1]
  нц для i от 2 до s
    ввод a
    макс[i] := max(a, макс[i - 1])
  кц
  | ввод остальных показаний, проверка пар
  вещ м | максимальное значение произведения

```

```

м := 0
нц для i от s + 1 до N
  ввод а
  макс[i] := max(a, макс[i - 1])
  м := max(м, а * макс[i - s])
кц
вывод м
кон

```

Обе приведённые программы эффективны по времени, но неэффективны по памяти: используемая память пропорциональна объёму исходных данных. Такие (и аналогичные по сути) программы оцениваются не выше 3 баллов.

Чтобы построить программу, эффективную по памяти, заметим, что, поскольку при обработке очередного элемента входных данных используется максимум, найденный на семь элементов раньше, достаточно хранить только семь последних максимумов. Весь алгоритм содержательно остаётся тем же, но нужно аккуратно организовать работу с массивом максимумов из семи элементов. Ниже приводится пример программы, реализующей эту идею

Программа 3. Пример правильной программы на алгоритмическом языке.

Программа эффективна и по времени, и по памяти

```

алг
нач
  цел s = 7 | требуемое расстояние между показаниями
  цел N
  ввод N
  вещ а | очередное показание прибора
  вещтаб макс[0:s - 1] | текущие максимумы
                       | последних 7 элементов

  цел i
  ввод макс[1]
  нц для i от 2 до s
    ввод а
    макс[mod(i, s)] := max(a, макс[i - 1])
  кц
  вещ м | максимальное значение произведения
  м := 0
  нц для i от s + 1 до N
    ввод а
    м := max(м, а * макс[mod(i, s)])
    макс[mod(i, s)] := max(a, макс[mod(i - 1, s)])
  кц
  вывод м
кон

```

Возможны и другие реализации этой идеи. Один из таких примеров приведён ниже. В этом случае в массиве длины 7 хранятся не максимумы, а исходные значения

Программа 4. Ещё один пример правильной программы на алгоритмическом языке. Программа эффективна и по времени, и по памяти

```

алг
нач
  цел s = 7 | требуемое расстояние между показаниями
  цел N
  ввод N
  вещтаб а[0:s - 1] | k-е введенное число
                  | записываем в ячейку а[mod(k, 7)]

```

```

вещ a_ | очередное показание прибора
цел i
| Пролог. Ввод первых семи чисел
нц для i от 1 до s
    ввод a_
    a[mod(i, s)] := a_
кц
| Ввод остальных значений,
| поиск максимального произведения
вещ макс = 0 | максимальное введенное число
                | (не считая 7 последних)
вещ м      | максимальное значение произведения
м := 0
нц для i от s + 1 до N
    ввод a_
    макс := max(макс, a[mod(i, s)])
    м := max(м, a_ * макс)
    a[mod(i, s)] := a_
кц
вывод м
кон

```

Ниже та же программа приведена на языке Паскаль

Программа 4а. Программа на языке Паскаль, эквивалентная программе 4

```

program c4;
const s = 7; {требуемое расстояние между показаниями}
var
  N: integer;
  a: array[0..s-1] of real; {хранение показаний прибора}
  {k-е введенное число записываем в ячейку a[k mod 7]}
  a_: real; {ввод очередного показания}
  mx: real; {максимальное введенное число}
  {(не считая 7 последних)}
  m: real; { максимальное значение произведения}
  i: integer;
begin
  readln(N);
  { Пролог. Ввод первых семи чисел}
  for i:=1 to s do
  begin
    readln(a_);
    a[i mod s] := a_
  end;
  { Ввод остальных значений, поиск максимального произведения}
  mx := 0; m := 0;
  for i := s + 1 to N do
  begin
    readln(a_);
    if a[i mod s] > mx then mx := a[i mod s];
    if a_ * mx > m then m := a_ * mx;
    a[i mod s] := a_
  end;
  writeln(m)
end.

```

К разделу 3. Информационные и коммуникационные технологии

3.1 4; 3.2 3; 3.3 1; 3.4 2; 3.5 3;

Разд. 3.2.1 4; Разд. 3.2.2 1; Разд. 3.2.3 4;

A8.1 2; A8.2 2; A8.3 3; A8.4 1; A8.5 4;

7.1 1; 7.2 3; 7.3 1; 7.4 3; 7.5 2;

7.6 5; 7.7 210; 7.8 10; 7.9 60; 7.10 5;

A6.1 4; A6.2 2; A6.3 4; A6.4 2; A6.5 2;

12.1 БВГА; 12.2 БВАЕЖГД; 12.3 CDFA; 12.4 СЕВА;

17.1 3214; 17.2 4123; 17.3 670; 17.4 4150; 17.5 20.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
СОДЕРЖАНИЕ КУРСА ИНФОРМАТИКИ И ИКТ В ТЕСТОВЫХ ЗАДАНИЯХ.....	7
1. Математические основы информатики	7
1.1. Кодирование информации	7
1.2. Системы счисления.....	19
1.3. Основы логики	21
1.4. Моделирование	29
2. Алгоритмизация и программирование	36
2.1. Исполнение алгоритмов	36
2.2. Программирование	62
2.3. Задания по программированию с развернутым ответом	91
3. Информационные и коммуникационные технологии.....	121
3.1. Файловые системы.....	121
3.2. Обработка графической информации.....	124
3.3. Цифровое кодирование звука	128
3.4. Обработка информации в электронных таблицах	130
3.5. Базы данных	137
3.6. Телекоммуникационные технологии.....	143
ОТВЕТЫ И РЕШЕНИЯ.....	150

