

ЕГЭ-тетрадь ИНФОРМАТИКА

Краткая теория

Формат реального ЕГЭ

Задания демо-версии 2011 года

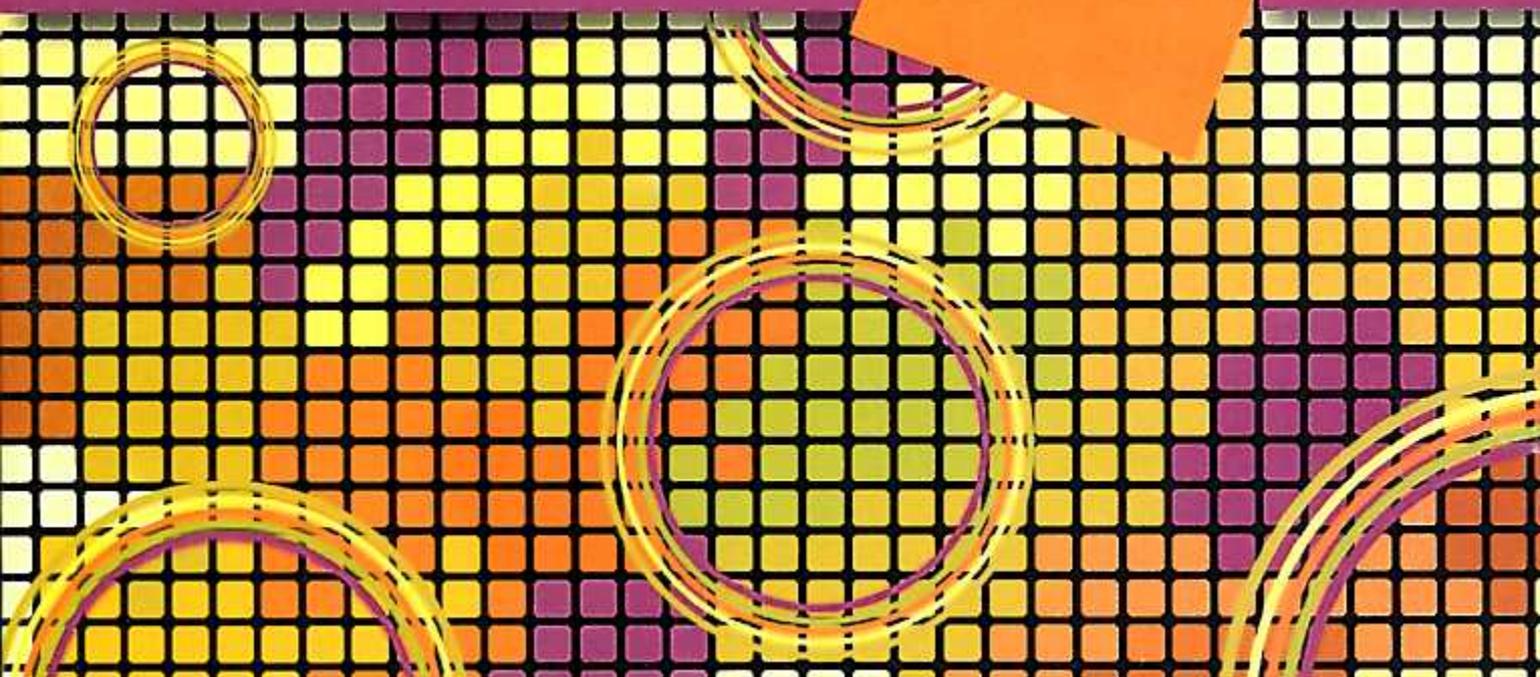
Задания экзаменов предыдущих лет

Задания, аналогичные экзаменационным

Раздел усложненных и комплексных задач

Разбор, решения, ответы

**ИНФОРМАТИКА И
ИНФОРМАЦИОННО-
КОММУНИКАЦИОННЫЕ
ТЕХНОЛОГИИ**



И. К. Сафронов

ЕГЭ-тетрадь ИНФОРМАТИКА

Санкт-Петербург
«БХВ-Петербург»

2011

УДК 681.3.06(075.3)
ББК 32.973.26я721
С12

Сафронов И. К.

С12 ЕГЭ-тетрадь. Информатика. — СПб.: БХВ-Петербург, 2011. — 184 с.: ил. — (ИиИКТ)
ISBN 978-5-9775-0621-2

В рабочей тетради для подготовки старшеклассников к сдаче ЕГЭ по информатике подробно разбираются задания демо-версии ЕГЭ-2011, ЕГЭ-2010 и заданий предыдущих лет, к ним приводятся аналогичные задачи. Экзаменационный материал представлен в соответствии с государственными образцами ЕГЭ по информатике, что позволяет готовиться по модели экзамена индивидуально. Есть раздел усложненных и комплексных задач. В конце тетради к большинству заданий приведены ответы.

Для образовательных учреждений

УДК 681.3.06(075.3)
ББК 32.973.26я721

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.12.10.
Формат 60×90^{1/8}. Печать офсетная. Усл. печ. л. 23.
Тираж 2000 экз. Заказ № 686
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

Предисловие

Ну что ж, уважаемый читатель, давайте познакомимся и поймем, что мы друг от друга хотим.

Я — Сафронов Игорь Константинович, преподаватель информатики в средних учебных заведениях со стажем уже 21 год. Автор нескольких книг по информатике, в том числе и по подготовке к ЕГЭ. Те, кого я лично готовил к экзамену, сдали его вполне благополучно, а некоторые даже отлично.

И вот из-под клавиатуры моего ноутбука родилась рабочая тетрадь для подготовки к ЕГЭ по информатике. Надеюсь, что она сослужит добрую пользу уважаемому читателю, если он поймет, что от него хотят авторы ЕГЭ, и *вдумчиво* прорешает все задания этой тетради. Тогда успех почти гарантирован!

Тетрадь устроена очень по-простому. Так как "отцы" ЕГЭ при подготовке рекомендуют опираться на их демо-версии, то я поступлю так: сначала иногда буду приводить краткую теорию (но совсем краткую — все же информатика в школе проходила), затем примеры заданий 2011 года (<http://www.fipi.ru/view/sections/217/docs/514.html>), а потом несколько самостоятельных заданий из предыдущих экзаменов и аналогичных для самостоятельного же решения. Глава с ответами тоже есть, но прибегайте к ней уж только чтобы себя проверить. Тем более, что на некоторые задания я ответы приберег.

Это иногда называют нехорошим словом "натаскивание". Пусть так, но у нас есть цель — сдать экзамен. И мы будем ВМЕСТЕ ее добиваться!

Итак, дорогой читатель, тебе предстоит сдать экзамен по информатике. Готов ли ты к этому — хорошо, а если нет — то давай я тебе немного помогу. Но без твоих усилий, конечно, не обойдется.

Все вопросы присылайте на адрес **old_matros@mail.ru**.

Ни пуха ни пера!

ГЛАВА 1

Кодирование текстовой информации, основные используемые кодировки

ТРЕБОВАНИЯ К ЧАСТЯМ А И В ЕГЭ

Учащиеся должны свободно оперировать различными единицами измерения объема информации, переходить от одних единиц к другим, представлять себе сравнительные объемы информации, системы компьютерного двоичного кодирования.

Краткие сведения о единицах измерения информационного объема и основных кодировках

◆ Единицы информационного объема:

1 бит — самая маленькая, неделимая единица — либо "0", либо "1".

1 байт = 8 бит.

1 Килобайт (Кб, Кбайт) = 1024 байта.

1 Мегабайт (Мб, Мбайт) = 1024 Кбайт = 2^{10} байт.

1 Гигабайт (Гб, Гбайт) = 1024 Мбайт = 2^{20} байт.

1 Терабайт (Тб, Тбайт) = 1024 Гбайт = 2^{30} байт.

◆ Основные кодировки.

ASCII — кодировка, в которой любой символ кодируется 8 битами (это касается и пробелов, и знаков препинания, и цифр).

KOI8-R (или КОИ-8) — то же самое, 8 бит на символ.

Unicode — кодировка, в которой любой символ кодируется 16 битами (или 2 байтами).

◆ Мощность алфавита.

Алфавитом называют любое упорядоченное множество знаков, используемых для обозначения чего-либо. Количество знаков в данном алфавите называется *мощностью алфавита*.

Примеры заданий с решениями

A2-2011

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной в 20 символов, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на

1) 320 бит; 2) 20 бит; 3) 160 байт; 4) 20 байт.

Решение.

Вычислим сначала первоначальный информационный объем сообщения:

16 бит на символ = 2 байта на символ, значит, $20 \times 2 = 40$ байт.

Теперь вычислим объем после перекодировки:

8 бит на символ = 1 байт на символ, значит $20 \times 1 = 20$ байт.

Найдем разность:

40 байт – 20 байт = 20 байт.

Правильный ответ: № 4.

A5-2011

Для передачи по каналу связи сообщения, состоящего только из символов А, Б, В и Г, используется неравномерный (по длине) код: А-00, Б-11, В-010, Г-011. Через канал связи передается сообщение: ГБВАВГ. Закодируйте сообщение данным кодом. Полученную двоичную последовательность переведите в шестнадцатеричную систему счисления. Какой вид будет иметь это сообщение?

1) 71013; 2) DBCACD; 3) 7A13; 4) 31A7.

Решение.

Ну что ж, закодируем сообщение предложенным кодом:

011 11 010 00 010 011.

Теперь разобьем код справа налево на шестнадцатеричные тетрады (см. разд. "Краткие сведения..." в главе 3). Получим:

0111 1010 0001 0011.

И заменим каждую тетраду соответствующей шестнадцатеричной цифрой:

7 A 1 3.

Правильный ответ: № 3.

ЗАМЕЧАНИЕ

Некоторые могут неосторожно вместо буквы А написать число 10 — и выбрать неправильный ответ № 1. Будьте бдительны!

A7-2011

Лена забыла пароль для входа в Windows XP, но помнила алгоритм его получения из символов "A153B42FB4" в строке подсказки. Если последовательность символов "B4" заменить на "B52" и из получившейся строки удалить все трехзначные числа, то полученная последовательность и будет паролем:

1) ABFB52; 2) AB42FB52; 3) ABFB4; 4) AB52FB.

Решение.

Не устаю удивляться, чего только не хотят проверить у выпускников! Для развития логического мышления в 3—4 классах замечательная задача, а у выпускников...

Тем не менее, не расслабляемся...

Заменяем в исходном коде все сочетания B4 на B52. (Американский бомбардировщик, что ли, составители имели в виду?)

A153B42FB4 \Rightarrow A153B522FB52.

Удалим из получившегося кода все трехзначные числа:

A153B522FB52 \Rightarrow ABFB52.

Правильный ответ: № 1.

ГЛАВА 2

Вычисление информационного объема

Требования и краткий курс см. в главе 1. Поэтому сразу перейдем к примерам 2011 года.

Примеры заданий с решениями

A16-2011

В некоторой стране автомобильный номер длиной 7 символов составляют из заглавных букв (используются только 22 различные буквы) 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 50 номеров.

- 1) 350 байт; 2) 300 байт; 3) 250 байт; 4) 200 байт.

Решение.

Итак, всего для номера могут использоваться 22 буквы и 10 цифр — всего 32 символа. Для кодирования каждого из 32 символов (по формуле Шеннона) необходимо минимально $2^5 = 32$ — 5 бит.

Тогда для семизначного номера потребуется $7 \times 5 = 35$ бит.

$35 \text{ бит} = 35 : 8 = 4,375$ байта, но по заданию — "каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым **ЦЕЛЫМ** количеством байт". И целым здесь будет не 4, а 5 байт. Надеюсь, понятно, почему?

Ну, а дальше проще некуда — для 50 номеров надо $50 \times 5 = 250$ байт.

Правильный ответ: № 3.

B6-2011

У Толи есть доступ к Интернету по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{19} бит в секунду. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Толи по низкоскоростному телефонному каналу со средней скоростью 2^{15} бит в секунду. Миша договорился с Толей, что тот будет скачивать для него данные объемом 5 Мбайт по высокоскоростному каналу и ретранслировать их Мише по низкоскоростному каналу.

Компьютер Толи 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 Толей данных до полного их получения Мишей?

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

ГЛАВА 3

Системы счисления

ТРЕБОВАНИЯ

Учащиеся должны знать позиционные и непозиционные системы счисления, их свойства, понятия о базе и основании систем счисления, свободно ориентироваться в десятичной, двоичной, восьмеричной, шестнадцатеричной системах счисления, переводить числа из одной системы в другую (пользуясь в том числе "триадами" и "тетрадами"), осуществлять простейшие арифметические действия над числами в разных системах счисления.

Краткие сведения систем счисления, применяемых в компьютерной технике

В компьютерной технике используются: двоичная, восьмеричная и шестнадцатеричная системы.

Восьмеричные триады:

"0" — 000 ₂	"4" — 100 ₂
"1" — 001 ₂	"5" — 101 ₂
"2" — 010 ₂	"6" — 110 ₂
"3" — 011 ₂	"7" — 111 ₂

Шестнадцатеричные тетрады:

"0" — 0000 ₂	"8" — 1000 ₂
"1" — 0001 ₂	"9" — 1001 ₂
"2" — 0010 ₂	"10 или A" — 1010 ₂
"3" — 0011 ₂	"11 или B" — 1011 ₂
"4" — 0100 ₂	"12 или C" — 1100 ₂
"5" — 0101 ₂	"13 или D" — 1101 ₂
"6" — 0110 ₂	"14 или E" — 1110 ₂
"7" — 0111 ₂	"15 или F" — 1111 ₂

Примеры заданий с решениями

A1-2011

Дано $A = A7_{16}$, $B = 251_8$. Какое из чисел C , записанных в двоичной системе, отвечает условию $A < C < B$?

- 1) 10101100₂;
- 2) 10101010₂;
- 3) 10101011₂;
- 4) 10101000₂.

Решение.

Ну, тут все просто. Переводим числа из шестнадцатеричной и восьмеричной систем счисления в двоичную через тетрады/триады:

$$A7_{16} = 1010\ 0111_2.$$

$$251_8 = 010\ 101\ 001_2 = 1010\ 1001_2.$$

И сразу видно, что между 0111 и 1001 лежит двоичное число 1000.

Правильный ответ: № 4.

A4-2011

Чему равна сумма чисел 57_8 и 46_{16} ?

1) 351_8 ; 2) 125_8 ; 3) 55_{16} ; 4) 75_{16} .

Решение.

Вариантов решения тут может быть несколько, в зависимости от того, насколько вы владеете навыками сложения в иных системах счисления, за исключением привычной десятичной.

1-й способ.

Переведем число 46_{16} в восьмеричную систему и сложим два числа. Применим тетрады.

$$46_{16} = 0100\ 0110_2 = 1\ 000\ 110_2 = 106_8.$$

$$57_8 + 116_8 = 165_8 = 001\ 110\ 101_2 = 0111\ 0101_2 = 75_{16}.$$

Правильный ответ: № 4.

2-й способ.

Переведем слагаемые в десятичную систему, потом сложим их и переведем ответ в восьмеричную и шестнадцатеричную системы.

$$57_8 = 5 \times 8^1 + 7 \times 8^0 = 47_{10}.$$

$$46_{16} = 4 \times 16^1 + 6 \times 16^0 = 70_{10}.$$

$$47_{10} + 70_{10} = 117_{10}.$$

$$117_{10} = 165_8 = 75_{16}.$$

Правильный ответ: № 4.

B4-2011

В системе счисления с некоторым основанием десятичное число 18 записывается в виде 30. Укажите это основание.

Решение.

Составляем простое уравнение, зная, как переводится число из любой системы счисления в десятичную:

$$30_p = 3 \times p^1 + 0 \times p^0 = 18_{10}.$$

$$3_p = 18.$$

$$p = 6.$$

Правильный ответ: 6.

B3-2009

Укажите через запятую в порядке возрастания все десятичные числа, *не превосходящие* 25, запись которых в системе счисления с основанием четыре оканчивается на 11.

Решение.

1-й способ. Как всегда — в лоб.

Возьмем и переведем все десятичные числа до 25 в систему с основанием 4 и выпишем подходящие по условию задания.

Напоминаю, что для перевода десятичного числа в любую другую систему счисления надо делить это число на основание системы счисления, в нашем случае на 4.

Путь долгий, но наградой будет правильный ответ. Мне его приводить здесь не хочется — уж больно долго.

Предлагаю взамен просто выписать все четверичные числа, соответствующие десятичным, от 1 до 25. Для этого надо знать, что в базе четверичной системы имеются только ЧЕТЫРЕ цифры — 0, 1, 2, 3. И больше ни-ни!

Десятичное число	Четверичное число
0	0
1	1
2	2
3	3

Далее проблема — у нас в десятичной системе идет число 4, а в четверичной системе такого числа нет — значит, подбираем такое ближайшее к 3 число, состоящее только из цифр четверичной системы. Это число 10. И так далее.

Десятичное число	Четверичное число	Десятичное число	Четверичное число
4	10	15	33
5	11	16	100
6	12	17	101
7	13	18	102
8	20	19	103
9	21	20	110
10	22	21	111
11	23	22	112
12	30	23	113
13	31	24	120
14	32	25	121

Правильный ответ: 5, 21.

2-й способ. Для думающих.

От нас требуют, чтобы мы нашли числа, которые заканчиваются на 11 в четверичной системе. Давайте эту самую систему выведем. Пойдем обратным путем. Какие числа в четверичной системе есть, такие, которые заканчиваются на 11?

Это, например, 11, 111, 211, 311. Пока хватит.

По очереди переведем их в десятичную систему и узнаем, укладываются ли они в заданный интервал от 1 до 25.

$$11_4 = 1 \times 4^1 + 1 \times 4^0 = 4 + 1 = 5_{10} \text{ — это нам подходит.}$$

$$111_4 = 1 \times 4^2 + 1 \times 4^1 + 1 \times 4^0 = 16 + 4 + 1 = 21_{10} \text{ — это нам тоже походит.}$$

ГЛАВА 4

Алгебра логики

ТРЕБОВАНИЯ

Учащиеся должны владеть понятиями элементарного высказывания значений True (Истина) и False (Ложь), знать, что такое инверсия (Логическое НЕ, отрицание), конъюнкция (Логическое И, логическое умножение), дизъюнкция (Логическое ИЛИ, логическое сложение), импликация (следование), таблицы их истинности, порядок логических действий, вычисление простых логических выражений, простейшие законы логики, представлять себе приложение законов логики как в повседневной жизни, так и в информатике.

ЗАМЕЧАНИЕ

В официальных заданиях ЕГЭ логическое отрицание НЕ обозначается символом "¬", например, ¬A, ¬(A & B). В разборе экзаменационных заданий я буду сохранять символ "¬", а в своих авторских заданиях буду использовать более распространенное и понятное обозначение: \bar{A} и A & B. Вы должны знать и то и другое обозначения.

Краткий курс алгебры логики

Таблица истинности для основных логических функций представлена в табл. 4.1.

Таблица 4.1

A	B	\bar{A} инверсия	$A \wedge B$ конъюнкция	$A \vee B$ дизъюнкция	$A \rightarrow B$ импликация
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

Приоритет логических операций:

1. Инверсия.
2. Конъюнкция.
3. Дизъюнкция.
4. Импликация.

Законы логики и правила равносильных преобразований логических выражений (многие из них похожи на арифметические законы):

1. *Переместительный (коммутативный) закон* (известен нам из математики — от перемены мест слагаемых сумма не меняется, от перемены мест сомножителей произведение не меняется):

$$X \vee Y = Y \vee X$$

$$X \wedge Y = Y \wedge X$$

2. Сочетательный (ассоциативный) закон

- ◆ для логического сложения:

$$(X \vee Y) \vee Z = X \vee (Y \vee Z)$$

- ◆ для логического умножения:

$$(X \wedge Y) \wedge Z = X \wedge (Y \wedge Z)$$

Вообще, когда знаки одинаковы (все \wedge или \vee), то можно скоб

3. Распределительный (дистрибутивный) закон

- ◆ для логического сложения:

$$(X \vee Y) \wedge Z = (X \wedge Z) \vee (Y \wedge Z)$$

- ◆ для логического умножения:

$$(X \wedge Y) \vee Z = (X \vee Z) \wedge (Y \vee Z)$$

Теперь законы чисто логические:

1. Закон двойной инверсии (двойного отрицания):

$$X = \overline{\overline{X}}$$

2. Закон общей инверсии (законы де Моргана)

- ◆ для логического сложения:

$$\overline{X \vee Y} = \overline{X} \wedge \overline{Y}$$

- ◆ для логического умножения:

$$\overline{X \wedge Y} = \overline{X} \vee \overline{Y}$$

3. Закон равносильности (идемпотентности):

$$X \vee X = X$$

$$X \wedge X = X$$

Из этого закона следует, что в логике степеней не бывает.

4. Законы исключения констант:

$$X \vee 1 = 1$$

$$X \vee 0 = X$$

$$X \wedge 1 = X$$

$$X \wedge 0 = 0$$

5. Закон противоречия:

$$X \wedge \overline{X} = 0$$

6. Закон исключения третьего:

$$X \vee \overline{X} = 1$$

7. Закон поглощения

- ◆ для логического сложения:

$$X \vee (X \wedge Y) = X$$

- ◆ для логического умножения:

$$X \wedge (X \vee Y) = X$$

8. Закон исключения (склеивания)

- ◆ для логического сложения:

$$(X \wedge Y) \vee (\overline{X} \wedge Y) = Y$$

- ◆ для логического умножения:

$$(X \vee Y) \wedge (\overline{X} \vee Y) = Y$$

9. Правило перевертывания (закон контрапозиции):

$$(X \rightarrow Y) = (Y \rightarrow X)$$

Примеры заданий с решениями

A9-2011

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	1	0
1	1	1	1
0	0	1	1

Какое выражение соответствует F ?

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

Решение.

Надо просто подставлять наборы переменных в предлагаемые функции и смотреть, что получается:

1. Пробуем первый набор: $X \wedge \neg Y \wedge \neg Z = 0 \wedge \neg 1 \wedge \neg 1 = 0$. Пока хорошо.
 Пробуем второй набор: $X \wedge \neg Y \wedge \neg Z = 1 \wedge \neg 1 \wedge \neg 1 = 0$. Уже плохо.
2. Пробуем первый набор: $\neg X \wedge \neg Y \wedge Z = \neg 0 \wedge \neg 1 \wedge 1 = 0$. Пока хорошо.
 Пробуем второй набор: $\neg X \wedge \neg Y \wedge Z = \neg 1 \wedge \neg 1 \wedge 1 = 0$. Опять плохо.
3. Пробуем первый набор: $\neg X \vee \neg Y \vee Z = \neg 0 \vee \neg 1 \vee 1 = 1$. Плохо сразу. Наверное, уж четвертый вариант правилен, но — проверить надо.
4. Пробуем первый набор: $X \vee \neg Y \vee \neg Z = 0 \vee \neg 1 \vee \neg 1 = 0$. Хорошо.
 Пробуем второй набор: $X \vee \neg Y \vee \neg Z = 1 \vee \neg 1 \vee \neg 0 = 1$. Опять хорошо.
 $X \vee \neg Y \vee \neg Z = 0 \vee \neg 0 \vee \neg 1 = 1$. Отлично!

Правильный ответ: № 4.

A10-2010

Укажите, какое логическое выражение равносильно выражению

$$A \vee \neg(\neg B \vee \neg C):$$

- 1) $\neg A \vee B \vee \neg C$; 3) $A \vee B \vee C$;
 2) $A \vee (B \wedge C)$; 4) $A \vee \neg B \vee \neg C$?

Решение.

Надо знать простые законы алгебры логики (см. разд. "Краткий курс алгебры логики" ранее в этой главе).

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

Правильный ответ: № 2.

A15-2011

Какое из приведенных имен удовлетворяет логическому условию:

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

- 1) ИРИНА;
2) АРТЕМ;

- 3) СТЕПАН;
4) МАРИЯ.

Решение.

Первая и вторая части логического выражения связаны конъюнкцией, что означает, что обе они для истинности всего выражения должны быть истинны. Вторая буква согласная у первого, второго и третьего имен. МАРИЯ отпала. Теперь левая часть. Тут вспоминаем, что импликация ложна в единственном случае: $1 \rightarrow 0$. То есть, последняя буква имени должна быть гласной. Сразу определилось правильное имя — ИРИНА. Тем более, что у этого имени действительно первая буква не согласная (И с чем она не согласная? Улыбнись, читатель!).

Правильный ответ: № 1.

B10-2011

Сколько различных решений имеет уравнение

$$((J \rightarrow K) \rightarrow (M \wedge N \wedge L)) \wedge ((J \wedge \neg K) \rightarrow \neg (M \wedge N \wedge L)) \wedge (M \rightarrow J) = 1,$$

где J, K, L, M, N — логические переменные?

В ответе *не нужно* перечислять все различные наборы значений J, K, L, M и N , при которых выполнено данное равенство. В качестве ответа нужно указать количество таких наборов.

Решение.

В данном случае надежнее подстраховаться и выписать все возможные наборы. Их $2^5 = 32$.

А потом подставлять эти наборы в данное уравнение и смотреть. По ходу подстановки какие-то варианты начнут массово отпадать. Например, когда J и K одновременно равны 0, то их импликация ИСТИННА, следовательно, при всех наборах L, M, N , в которых есть хотя бы один 0, импликация $(J \rightarrow K) \rightarrow (M \wedge N \wedge L)$ будет ЛОЖНА, а так как она является составляющей конъюнкции, то и она будет ЛОЖНА и сразу несколько вариантов отпадают.

Если с такой сложной логикой есть проблемы — методично подставьте все комбинации. Только внимательно!

Таблица 4.2

J	K	L	M	N	Уравнение	J	K	L	M	N	Уравнение
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	1	0	0	0	1	0
0	0	0	1	0	0	1	0	0	1	0	0
0	0	0	1	1	0	1	0	0	1	1	0
0	0	1	0	0	0	1	0	1	0	0	0
0	0	1	0	1	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0	1	1	0	0
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	0	1	1	1	0	0	0	0
0	1	0	0	1	1	1	1	0	0	1	0

ГЛАВА 5

Электронные таблицы и диаграммы

ТРЕБОВАНИЯ

Учащиеся должны знать основы работы в электронных таблицах. Типы данных. Диапазоны ячеек. Абсолютную, относительную и смешанную адресации. Формулы и основные функции. Построение графиков и диаграмм. Анализ готовых таблицы и диаграмм.

Примеры заданий с решениями

A11-2011

В динамической (электронной) таблице приведены значения посевных площадей (в га) и урожая (в центнерах) четырех зерновых культур в четырех хозяйствах одного района. В каком из хозяйств достигнута максимальная урожайность зерновых (по валовому сбору)? (Урожайность измеряется в центнерах с гектара.)

Зерновые культуры	Название хозяйства							
	Заря		Первомайское		Победа		Рассвет	
	Посевы	Урожай	Посевы	Урожай	Посевы	Урожай	Посевы	Урожай
Пшеница	600	15 300	900	23 800	300	7500	1200	31 200
Рожь	100	2150	500	12 000	50	1100	250	5500
Овес	100	2350	400	10 000	50	1200	200	4800
Ячмень	200	6000	200	6300	100	3100	350	10 500
<i>Всего зерновые</i>	<i>1000</i>	<i>25 800</i>	<i>2000</i>	<i>52 100</i>	<i>500</i>	<i>12 900</i>	<i>2000</i>	<i>52 000</i>

- 1) Заря;
- 2) Первомайское;
- 3) Победа;
- 4) Рассвет.

Решение.

Ну, вот это вообще сильный ход со стороны разработчиков тестов. Мало того, что экзамен принимается на бумаге, так еще и калькуляторами пользоваться нельзя. И проверяется на экзамене по информатике что? Правильно, умение считать на бумажке в столбик. Умение нужное, но на информатике ли его надо проверять?

Вопросы, конечно, риторические. А нам экзамен надо сдавать.

Итак, считаем (я все же воспользуюсь калькулятором):

- 1) "Заря" — $25800 : 1000 = 25,8$ центнера с гектара;
- 2) "Первомайское" — $52100 : 2000 = 26,05$ центнера с гектара;
- 3) "Победа" — $12900 : 500 = 25,8$ центнера с гектара;
- 4) "Рассвет" — $52000 : 2000 = 26$ центнеров с гектара.

Молодцы в "Первомайском"!

Правильный ответ: № 2.

A12-2011

Торговое предприятие владеет тремя магазинами (I, II и III), каждый из которых реализует периферийные компьютерные устройства: мониторы (М), принтеры (П), сканеры (С) или клавиатуры (К). На диаграмме 1 (рис. 5.1) показано количество проданных товаров каждого вида за месяц. На диаграмме 2 показано, как за тот же период соотносятся продажи товаров (в штуках) в трех магазинах предприятия.

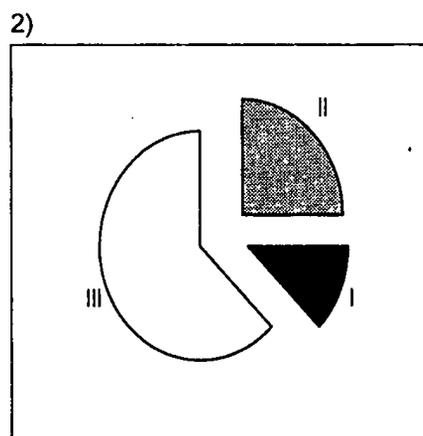
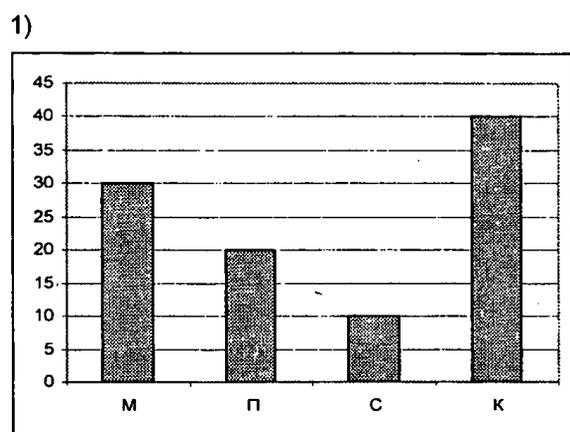


Рис. 5.1

Какое из приведенных ниже утверждений следует из анализа обеих диаграмм?

- А) Все сканеры могли быть проданы через магазин III.
- Б) Все принтеры и сканеры могли быть проданы через магазин II.
- В) Все мониторы могли быть проданы через магазин I.
- Г) Ни один принтер не был продан через магазин II.

- 1) А;
- 2) Б;
- 3) В;
- 4) Г.

Решение.

Сначала проанализируем количество проданного товара по первой диаграмме:

- ◆ мониторы — 30%;
- ◆ принтеры — 20%;
- ◆ сканеры — 10%;
- ◆ клавиатуры — 40%.

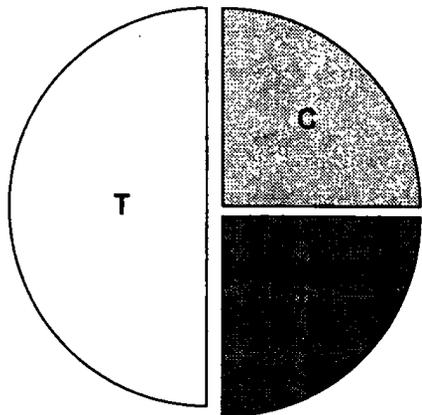
Решение.

Правильный ответ																							

A17-2010

В цехе трудятся рабочие трех специальностей: токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд не меньший второго и не больший пятого. На диаграмме 1 (рис. 5.3) отражено распределение рабочих по специальностям, а на диаграмме 2 — количество рабочих с различными разрядами. Каждый рабочий имеет только одну специальность и один разряд.

1)



2)

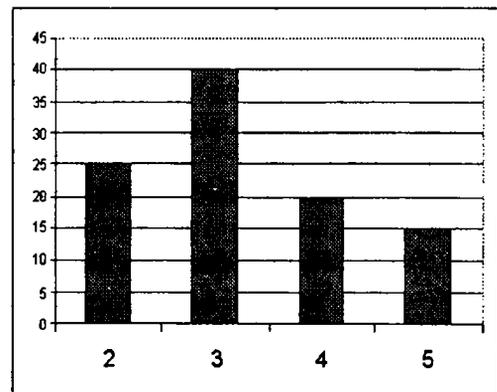


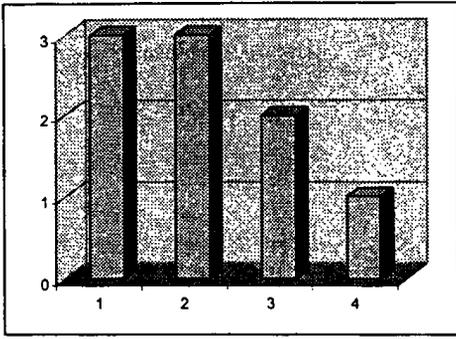
Рис. 5.3

Какое из утверждений:

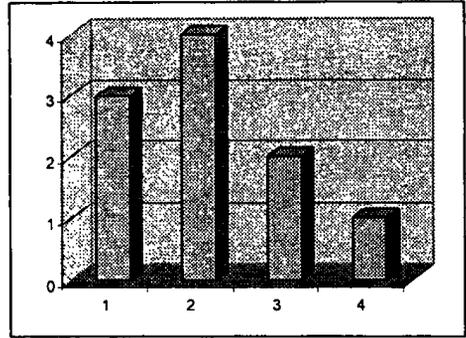
- А) среди слесарей найдется хотя бы один третьего разряда;
 - Б) среди токарей найдется хотя бы один второго разряда;
 - В) все токари могут иметь четвертый разряд;
 - Г) все фрезеровщики могут иметь третий разряд
- следует из диаграмм?

- 1) А;
- 2) Б;
- 3) В;
- 4) Г.

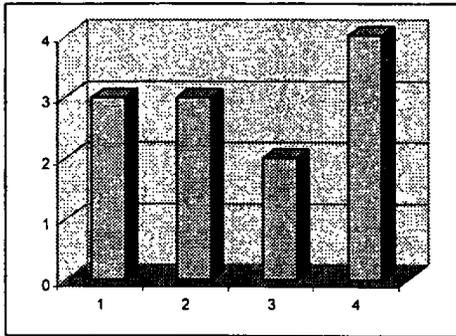
1)



3)



2)



4)

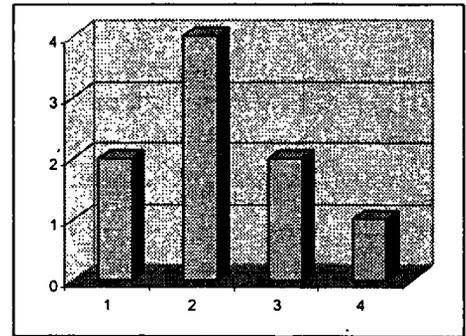


Рис. 5.4

ГЛАВА 6

Закономерности¹

ТРЕБОВАНИЯ

Учащиеся должны владеть элементарной логикой и навыками выполнения арифметических действий.

Примеры заданий с решениями

V8-2011

Строки (цепочки символов латинских букв) создаются по следующему правилу.

Первая строка состоит из одного символа — латинской буквы "A". Каждая из последующих цепочек создается такими действиями: в очередную строку сначала записывается буква, чей порядковый номер в алфавите соответствует номеру строки (на i -м шаге пишется i -я буква алфавита), к ней слева дважды подряд приписывается предыдущая строка.

Вот первые 4 строки, созданные по этому правилу:

- (1) A
- (2) AAB
- (3) AABAAB
- (4) AABAABCAABAABCD

Латинский алфавит (для справки):

ABCDEFGHIJKLMNOPQRSTUVWXYZ.

Имеется задание: определить символ, стоящий в n -й строке на позиции $2^{n-1} - 5$, считая от левого края цепочки.

Выполните это задание для $n = 8$.

Решение.

Сначала определим закономерность количества символов в строке. В первой — 1, во второй — 3, в третьей — 7, в четвертой — 15. Вывод простой $S = 2^n - 1$, где S — количество символов в строке, n — номер строки.

Теперь можем определить, сколько символов в искомой строке. $S_8 = 2^8 - 1 = 255$.

Далее, из условия следует, что в конце восьмой строки будет уже восемь подряд идущих букв латинского алфавита, а именно ABCDEFGH. А слева от этого восьмибуквенного сочетания записана дважды предыдущая строка, седьмая, в которой было 127 символов.

Вычислим позицию символа, который требуется определить.

$$2^{n-1} - 5 = 2^7 - 5 = 123.$$

¹ По странной прихоти отцов ЕГЭ — почти исключительно цепочки или иные... бусины.

Теперь все просто. В седьмой строке (состоящей из 127 символов) в конце 7 символов латинского алфавита:

Номер символа	121	122	123	124	125	126	127
Символ	A	B	C	D	E	F	G

Правильный ответ: C.

Решаем самостоятельно

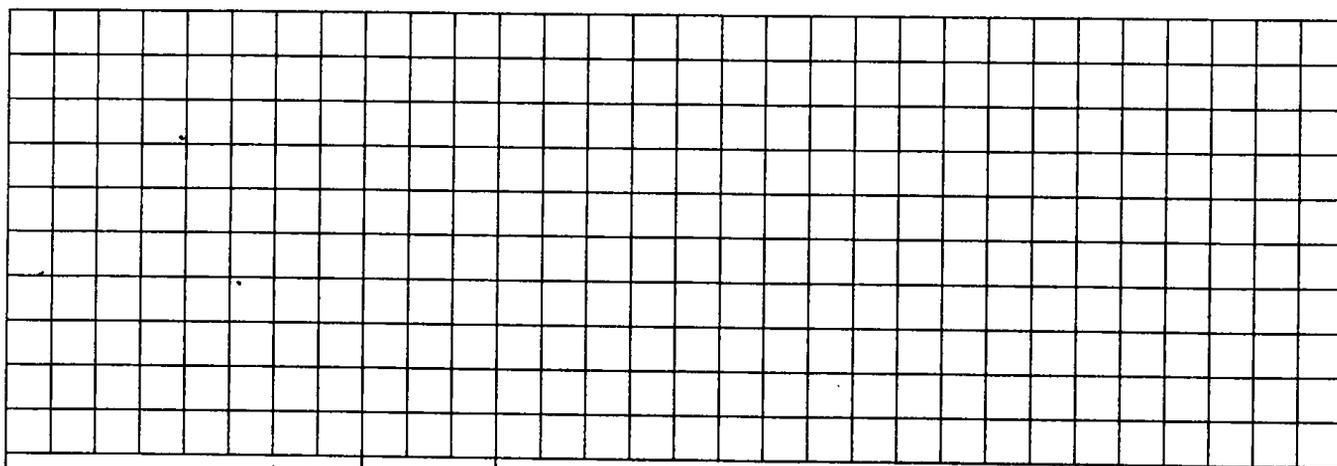
A12-2009

Цепочка из трех бусин, помеченных латинскими буквами, формируется по следующему правилу. В конце цепочки стоит одна из бусин A , B , C . На первом месте — одна из бусин B , D , C , которой нет на третьем месте. В середине — одна из бусин A , C , E , B , не стоящая на первом месте.

Какая из перечисленных цепочек создана по этому правилу?

- 1) CBV ; 3) BCD ;
2) EAC ; 4) BCB .

Решение.



Правильный ответ

B6-2008

Записано 7 строк, каждая имеет свой индекс — от 0 до 6. В нулевой строке записана цифра 0 (ноль).

Каждая строка состоит из двух повторений предыдущей и добавленного в конец своего индекса (в i -й строке в конце добавлена цифра i).

Вот первые 4 строки, созданные по этому правилу:

- (0) 0
(1) 001
(2) 0010012
(3) 001001200100123

Какая цифра стоит в последней строке на 124-м месте (считая слева направо)?

ГЛАВА 7

Базы данных

Примеры заданий с решениями

A6-2011

Путешественник пришел в 08:00 на автостанцию населенного пункта ЛИСЬЕ и обнаружил следующее расписание автобусов для всей районной сети маршрутов:

Пункт отправления	Пункт прибытия	Время отправления	Время прибытия
ЛИСЬЕ	ЗАЙЦЕВО	07:50	09:05
СОБОЛЕВО	ЛИСЬЕ	08:55	10:05
ЕЖОВО	ЛИСЬЕ	09:05	10:15
ЗАЙЦЕВО	ЕЖОВО	10:00	11:10
ЛИСЬЕ	СОБОЛЕВО	10:15	11:30
ЛИСЬЕ	ЕЖОВО	10:45	12:00
ЗАЙЦЕВО	ЛИСЬЕ	11:05	12:15
СОБОЛЕВО	ЗАЙЦЕВО	11:10	12:25
ЕЖОВО	ЗАЙЦЕВО	12:15	13:25
ЗАЙЦЕВО	СОБОЛЕВО	12:45	13:55

Определите самое раннее время, когда путешественник сможет оказаться в пункте ЗАЙЦЕВО согласно этому расписанию.

- 1) 09:05;
- 2) 12:15;
- 3) 12:25;
- 4) 13:25.

Решение.

Рассмотрим все возможные варианты передвижения путешественника.

1. ЛИСЬЕ — ЗАЙЦЕВО напрямую. Кажется бы, вот радость — прибытие в 09:05, и задача решена. Но наш путешественник опоздал на 10 минут, автобус ушел...
2. ЛИСЬЕ — СОБОЛЕВО — ЗАЙЦЕВО. В СОБОЛЕВО он будет 11:30, и... опять опаздывает на автобус...
3. ЛИСЬЕ — ЕЖОВО — ЗАЙЦЕВО. В ЕЖОВО он попадет 10.45, успеет на автобус из ЕЖОВО в 12.15 и уже в 13.25 будет на месте. Ура!

Правильный ответ: № 4.

A13-2011

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, пр. 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 г.
002	D02	01002	100	5/01/2009 г.
003	D06	01002	200	5/01/2009 г.
004	D01	02002	20	5/01/2009 г.
005	D02	02002	30	5/01/2009 г.
006	D02	01003	20	6/01/2009 г.

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.

ГЛАВА 8

Файловая структура организации информации на компьютере

ТРЕБОВАНИЯ

Учащиеся должны владеть знаниями об организации информации, используемой в компьютере, понятиями об именах дисков, файлов, каталогов (папок), подкаталогов, их вложенности друг в друга.

Кроме того, хорошо бы представлять себе пути к файлам, последовательность открываемых при этом подкаталогов. (Кто изучал DOS, тому проще.)

Примеры заданий с решениями

A3-2011

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:

1234.xls

23.xml

234.xls

23.xml

1) *23*.?x*

2) ?23?.x??

3) ?23?.x*

4) *23*.???

Решение.

Вся теория этого задания изложена в самом задании. Поэтому можно было бы информатику вовсе не изучать, а задание, тем не менее, успешно выполнить. Сделаем?

В указанной группе у всех имен файлов присутствует число 23. В двух именах перед этим числом ничего нет, значит, ответы 2 и 3 отпадают (т. к. наличие в маске вопросительного знака означает ОБЯЗАТЕЛЬНОЕ присутствие символа). Остались варианты 1 и 4. Вариант 1 отпадает по причине обязательного символа перед буквой "x" в расширении. Остается ответ 4, где в расширении должно быть 3 символа.

Правильный ответ: № 4.

ГЛАВА 9

RGB-кодирование

ТРЕБОВАНИЯ

Учащиеся должны иметь четкое представление о цветовых схемах компьютера. Знать шестнадцатеричное кодирование цветов. Уметь по шестизначному RGB-коду представить цвет.

Краткие сведения об основных цветах при шестнадцатеричном RGB-кодировании

- ◆ #000000 — черный.
- ◆ #FFFFFF — белый.
- ◆ #FF0000 — красный.
- ◆ #00FF00 — зеленый.
- ◆ #0000FF — синий.
- ◆ #FFFF00 — желтый.
- ◆ #00FFFF — голубой (а если добавить #80 красного — неожиданно получится оранжевый).
- ◆ #FF00FF — фиолетовый (пурпурный).

Если же все цвета в равных пропорциях, то получаются оттенки серого...

Примеры заданий с решениями

A14-2011

Для кодирования цвета фона интернет-страницы используется атрибут `b bgcolor="#xxxxxx"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонентов в 24-битной RGB-модели следующим образом:

XX	XX	XX
красный	зеленый	синий

К какому цвету будет близок цвет страницы, заданный тегом `<body bgcolor="#747474">?`

- 1) серый;
- 2) белый;
- 3) фиолетовый;
- 4) черный.

Решение.

Если все компоненты нули (минимум), то это будет черный, если все FF (максимум), то белый, а если они в этом промежутке и все одинаковы — то это все оттенки серого, в данном случае светло-серого.

Правильный ответ: № 1.

ГЛАВА 10

Исполнители

ТРЕБОВАНИЯ

Учащиеся должны знать типы, свойства алгоритмов. Уметь четко и однозначно исполнять предложенные алгоритмы. Почувствовать себя роботом. Анализировать поставленные задачи и делать выводы.

Примеры заданий с решениями

A18-2011

Система команд исполнителя "РОБОТ", "живущего" в прямоугольном лабиринте на клетчатой плоскости, такова:

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

При выполнении этих команд "РОБОТ" перемещается на одну клетку соответственно: вверх \uparrow , вниз \downarrow , влево \leftarrow , вправо \rightarrow .

Четыре команды проверяют истинность условия отсутствия стены у 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 (рис. 10.1) 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) 4.

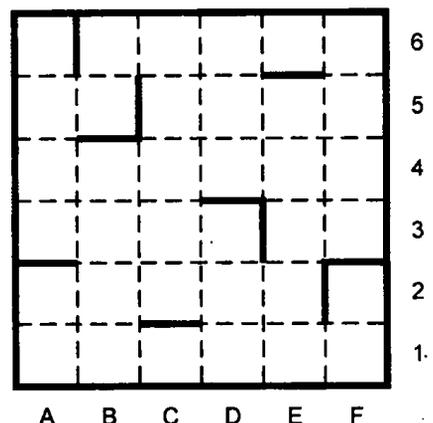


Рис. 10.1

Решение.

Пути два — один длинный, но надежный и думать не сильно нужно: просто надо побыть очень хорошим исполнителем и не ошибиться при выполнении команд.

По очереди выбираем клетки поверхности для старта и смотрим, куда придем в результате выполнения.

Например, А6. Справа несвободно, поэтому никуда не двигаемся, переходим ко второму условию. Снизу свободно, начинаем двигаться влево и... разбиваемся о стену. А1 — смертельно!

В6. Справа свободно, движемся вниз, в В5 становится справа несвободно — переходим к следующему условию. Снизу несвободно, остаемся в В5. Следующее условие — слева свободно, движемся вверх и тут же, в В6, слева перестает быть свободно, да и с последним условием нехорошо, остановились, т. е. вернулись в В6. Ура! Первая нашлась!

И т. д. После 36 исполнений выяснится, что искомым клеткам три, а именно В6, Е5 и D3.

Можно, правда, подумать и немножко сократить решение. Так как последняя команда проверяет условие ПОКА <сверху свободно>, то чтобы "РОБОТ" остановился, надо чтобы сверху НЕ БЫЛО свободно, т. е. обязательно должна быть граница сверху. И вот уже из 36 остается для проверки только 11! Экономим время, 25 клеток проверять НЕ НУЖНО!

Правильный ответ: № 3.

V3-2011

У исполнителя "Калькулятор" две команды, которым присвоены номера:

1. прибавь 1.
2. умножь на 3.

Выполняя первую из них, "Калькулятор" прибавляет к числу на экране 1, а выполняя вторую, утраивает его. Запишите порядок команд в программе получения из 2 числа 26, содержащей не более 6 команд, указывая лишь номера команд.

(Например, программа 21211 — это программа

умножь на 3

прибавь 1

умножь на 3

прибавь 1

прибавь 1

которая преобразует число 1 в 14.)

Решение.

Проанализируем с конца, а как, собственно, может получиться из числа 2 число 26 на таких условиях? Из 13 не может — нет умножения на 2. Может из 24 двойным прибавлением единицы. 24, в свою очередь, получается утраиванием 8, восемь — двойным прибавлением единицы из 6, а уж 6 — утраиванием изначальной двойки. Вот и решение: 211211.

Правильный ответ: 211211.

Решаем самостоятельно

V5-2009

У исполнителя "Калькулятор" две команды, которым присвоены номера:

1. прибавь 3.
2. умножь на 4.

Выполняя первую из них, "Калькулятор" прибавляет к числу на экране 3, а выполняя вторую, умножает его на 4. Запишите порядок команд в про-

Задание 10.5

Исполнитель считывает символ с входной ленты и помещает его в память. Начиная со второго символа, исполнитель сравнивает считанный символ с символом, хранящимся в памяти. В случае их совпадения на выходную ленту записывается 1, иначе — 0, после чего считанный символ помещается в память вместо прежнего. Запишите последовательность символов, которая получится на выходной ленте после обработки строки:

SSSWWSWW

Решение.

Правильный ответ																															

ГЛАВА 11

Алгоритмизация

ТРЕБОВАНИЯ

Учащиеся должны иметь представление о блок-схемах и их обозначениях, разбираться в ветвлениях и циклах. Представлять себе работу с массивами в цикле хотя бы в одном из основных языков программирования, предлагаемых к изучению в школе.

Примеры заданий с решениями

A8-2011

Определите значение переменной c после выполнения следующего фрагмента программы, в котором a , b и c — переменные вещественного (действительного) типа.

Бейсик

```
a = 120
b = 100
a = a + b / 2
IF b < a / 2 THEN
c = b + a
ELSE
c = b + a / 2
ENDIF
```

Паскаль

```
a := 120;
b := 100;
a := a + b / 2;
if b < a / 2
then
  c := b + a
else
  c := b + a / 2;
```

Си

```
a = 120;
b = 100;
a = a + b / 2;
if (b < a / 2)
  c = b + a;
else
  c = b + a / 2;
```

Алгоритмический язык

```
a := 120
b := 100
a := a + b / 2
если b < a / 2
то c := b + a
иначе c := b + a / 2
все
```

- 1) $c = 105$;
- 2) $c = 160$;
- 3) $c = 185$;
- 4) $c = 270$.

Решение.

Все очень просто. После присвоения исходных значений переменным a и b в третьей строке программы вычисляется новое значение переменной a . Подставляем:

$$a = 120 + 100/2 = 120 + 50 = 170.$$

Теперь проверяем условие: если b (равное 100) меньше $a/2$ ($170/2 = 85$), то c становится равным сумме a и b , иначе c становится равным сумме b и половины a , т. е. в нашем случае, $100 + 85 = 185$.

Правильный ответ: № 3.

A17-2011

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

Бейсик

```
FOR i=0 TO 10
A(i)=i-1
NEXT i
FOR i=10 TO 1 STEP -1
A(i-1)=A(i)
NEXT i
```

Паскаль

```
for i:=0 to 10 do
  A[i]:=i-1;
for i:=10 downto 1 do
  A[i-1]:=A[i];
```

Си

```
for (i=0;i<=10;i++)
A[i]=i-1;
for (i=10;i>=1;i--)
  A[i-1]=A[i];
```

Алгоритмический язык

```
нц для i от 0 до 10
  A[i]:=i-1
кц
нц для i от 10 до 1 шаг -1
  A[i-1]:=A[i]
кц
```

Чему окажутся равны элементы этого массива?

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

Решение.

Ну что, циклов два. Оба — типа FOR, с заранее известным количеством повторений. В нашем случае от 0 до 10 — цикл повторяется 11 раз.

В первом происходит заполнение массива A — на места с нулевого по десятое записываются значения с -1 до 9.

А вот второй цикл покрутим:

Номер шага	Значение i	Значения $A(i-1), A(i)$
1	10	$A(9) = A(10) = 9$
2	9	$A(8) = A(9) = 9$
3	8	$A(7) = A(8) = 9$
4	7	$A(6) = A(7) = 9$
5	6	$A(5) = A(6) = 9$
6	5	$A(4) = A(5) = 9$
7	4	$A(3) = A(4) = 9$
8	3	$A(2) = A(3) = 9$
9	2	$A(1) = A(2) = 9$
10	1	$A(0) = A(1) = 9$

Ну а $A(10)$ так и осталось равно 9.

Правильный ответ: № 1.

B2-2011

Запишите значение переменной a после выполнения фрагмента алгоритма (рис. 11.1).

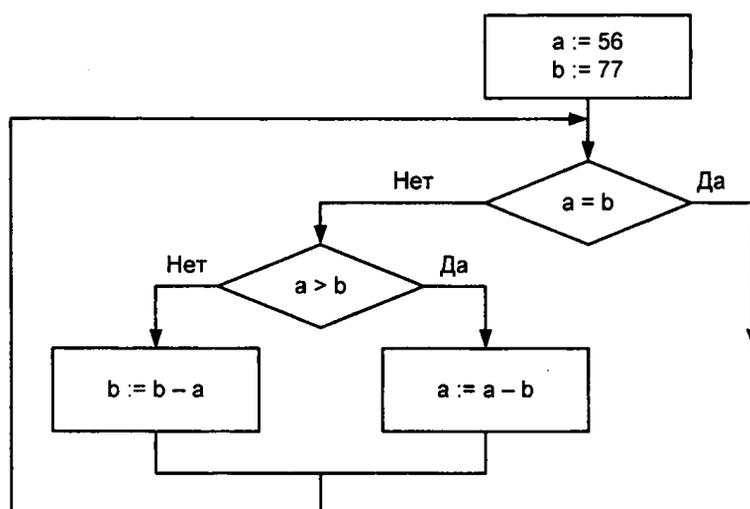


Рис. 11.1

Примечание: знаком $:=$ обозначена операция присваивания.
В бланк ответов впишите только число.

Решение.

Ничего необычного — вновь знай, исполняй.

Си

Алгоритмический язык

```

for (i = 0; i <= 10; i++)      нц для i от 0 до 10
    A[i] = i;                  A[i] := i
for (i = 0; i <= 10; i++)      кц
{                                нц для i от 0 до 10
    A[10-i] = A[i];           A[10-i] := A[i]
    A[i] = A[10-i];          A[i] := A[10-i]
}                                кц
    
```

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.

Правильный ответ																						

B2-2010

Запишите значение переменной b после выполнения фрагмента алгоритма (рис. 11.2).

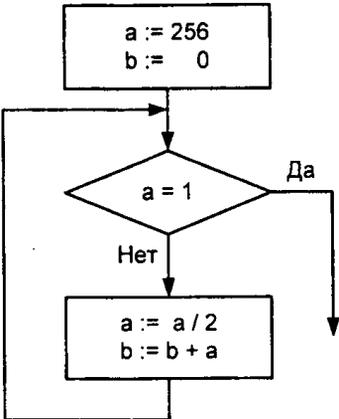


Рис. 11.2

Задание 11.2

В программе описан одномерный целочисленный массив А с индексами от 1 до 10. Ниже представлен фрагмент одной и той же программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются:

Бейсик	Паскаль
FOR i = 1 TO 10	for i := 1 to 10 do
A(i) = i - 1	A[i] := i - 1;
NEXT i	for i := 1 to 10 do
FOR i = 1 TO 10	A[i] := A[11-i];
A(i) = A(11-i)	
NEXT i	

Как будет выглядеть массив после изменения значений его элементов?

- 1) 9 8 7 6 5 4 3 2 1 0;
- 2) 1 2 3 4 5 5 4 3 2 1;
- 3) 11 10 9 8 7 6 5 4 3 2;
- 4) 9 8 7 6 5 5 6 7 8 9.

Решение.

Правильный ответ																				

Задание 11.3

Дан фрагмент программы, обрабатывающей массив А из n элементов, причем известно, что A[1] отлично от нуля:

Бейсик	Паскаль
pr = 1	pr := 1;
i = 1	i := 1;
WHILE (i <= n) AND ((A(i) < 0) OR (A(i) > 0))	while(i <= n) and ((A[i] < 0) or (A[i] > 0)) do
pr = pr * A(i)	begin
i = i + 1	pr := pr * A[i];
WEND	i := i + 1
	end

После выполнения данного алгоритма значение переменной pr будет равно произведению:

- 1) всех ненулевых элементов массива A;
- 2) всех положительных элементов массива A;
- 3) всех отрицательных элементов массива A;
- 4) ненулевых элементов массива A от A[1] до первого нулевого элемента.

Решение.

<i>Правильный ответ</i>	

Задание 11.4

Дан фрагмент программы, обрабатывающей массив A из n элементов:

<pre> Бейсик J = 1 FOR I = 1 TO N IF A(I) > A(J) THEN J = I NEXT I m = J </pre>	<pre> Паскаль j := 1; for i := 1 to n do if A[i] > A[j] then j := i; m := j; </pre>
--	--

Чему будет равно значение переменной m после выполнения данного алгоритма?

- 1) Максимальному элементу в массиве A.
- 2) Номеру максимального элемента в массиве A (первому из них, если максимальных элементов несколько).
- 3) Номеру максимального элемента в массиве A (последнему из них, если максимальных элементов несколько).
- 4) Количеству элементов, равных максимальному в массиве A.

Решение.

<i>Правильный ответ</i>	

Задание 11.8

Что будет выведено на экран в результате выполнения фрагмента следующей программы?

Бейсик	Паскаль
X = 13 Y = 52 Z = 99 FOR U = 100 TO 1 STEP -2 IF U = X THEN PRINT U IF U = Y THEN PRINT U IF U = Z THEN PRINT U NEXT U	x := 13; y := 52; z := 99; u := 100; 1: if u = x then writeln(u); if u = y then writeln(u); if u = z then writeln(u); u := u - 2; if u > 1 then goto 1;

- | | | | |
|-------|-------|-------|-------|
| 1) 13 | 2) 13 | 3) 52 | 4) 52 |
| 99 | 52 | | -2 |
| | 99 | | |

Решение.

Правильный ответ

Задание 11.9

Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

Бейсик	Паскаль
a = 2007 b = (a \ 100) + 100 a = b \ 10 - a MOD 1000	a := 2007; b := (a div 100)+100; a := b div 10 - a mod 1000;

(\ и MOD – операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно)

{div и mod – операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}

ГЛАВА 12

Логические задачи

ТРЕБОВАНИЯ

Учащиеся должны уметь логически мыслить (что, вообще-то, должно быть свойственно любому человеку разумному, а не только сдающему ЕГЭ по информатике).

Примеры заданий с решениями

В7-2011

Девять школьников, оставшихся в классе на перемене, были вызваны к директору. Один из них разбил окно в кабинете. На вопрос директора, кто это сделал, были получены следующие ответы:

- 1) Володя: "Это сделал Саша";
- 2) Аня: "Володя лжет!";
- 3) Егор: "Маша разбила";
- 4) Саша: "Аня говорит неправду!";
- 5) Рома: "Разбила либо Маша, либо Нина...";
- 6) Маша: "Это я разбила!";
- 7) Нина: "Маша не разбивала!";
- 8) Коля: "Ни Маша, ни Нина этого не делали";
- 9) Олег: "Нина не разбивала!".

Кто разбил окно, если известно, что из этих девяти высказываний истинны только три?

Ответ запишите в виде первой буквы имени.

Решение.

Ну что ж, вероятность просто угадать — всего лишь одна девятая. Не густо. Придется подумать и повесить свои шансы.

Попробуем искать поочередно, какие из высказываний могли бы быть истинными, и при этом их было бы ровно 3, и они не противоречили бы остальным.

Начнем с прямых высказываний о виновниках.

1. Допустим, Володя сказал правду, и стекло разбил Саша. Тогда: Аня — лжет, Егор — лжет, Саша — правдив, Рома лжет, Маша лжет, Нина правдива, Коля правдив. СТОП. Уже четыре ПРАВДЫ — противоречие, однако.
2. Допустим, Егор говорит правду, и стекло разбила Маша. Тогда: Володя лжет, Аня говорит правду, Саша лжет, Рома — правдив, Маша правдива... Опять уже четыре ПРАВДЫ!
3. Допустим, прав Рома, и стекло разбила либо Маша, либо Нина. Но Маши уже предыдущим пунктом обвинения сняты. Значит, предполагаем, что стекло разбила Нина. Тогда: Володя лжет, Аня правдива, Егор лжет, Саша лжет, Маша лжет, Нина правдива, Коля лжет, Олег лжет. И у нас ровно ТРИ ПРАВДЫ-ИСТИНЫ! Ура! Нина разбила стекло, конечно, но оказалась честной девушкой.

Правильный ответ: Н.

ГЛАВА 13

Интернет

ТРЕБОВАНИЯ

Учащиеся должны знать адресацию в Интернете, основные теги HTML, принципы работы поисковых машин, правила поиска информации во Всемирной сети.

Примеры заданий с решениями

В4-2011

На месте преступления были обнаружены четыре обрывка бумаги (рис. 13.1). Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами "А", "Б", "В" и "Г". Восстановите IP-адрес.

В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

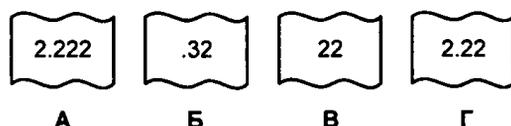


Рис. 13.1

Решение.

Фрагмент "Б" не может стоять вначале (начинается с точки) и не может в середине, т. к. никакой другой фрагмент не может его продолжить. Значит, он последний. Фрагмент "А", в свою очередь, не может быть непосредственно перед фрагментами "В" и "Г" — а то получится ужас типа 2.22222 или 2.2222.22. Значит, фрагмент "А" был третьим. И у нас уже есть X.X2.222.32. Осталось расположить фрагменты "В" и "Г". Вариантов всего два:

- ♦ пробуем ВГАБ — получаем 222.222.222.32 — годится;
- ♦ пробуем ГВАБ — получаем 2.22222.222.32 — не годится.

Правильный ответ: ВГАБ.

В9-2011

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 Интернета.

ГЛАВА 14

Нахождение ошибок в готовых программах и их доработка

Примеры заданий с решениями

C1-2011

Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки (рис. 14.1) заданной заштрихованной области (включая границы). Программист торопился и написал программу неправильно.

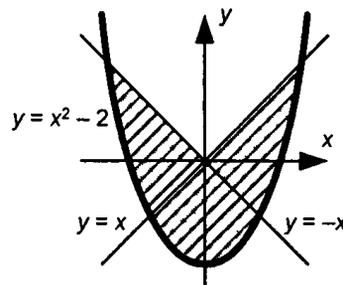


Рис. 14.1

Паскаль

```
var x,y: real;
begin
readln(x,y);
if y<=x then
if y<=-x then
if y>=x*x-2 then
  write('принадлежит')
else
  write('не принадлежит')
end.
```

Бейсик

```
INPUT x, y
IF y<=x THEN
IF y<=-x THEN
IF y>=x*x-2 THEN
PRINT "принадлежит"
ELSE
PRINT "не принадлежит"
ENDIF
ENDIF
ENDIF
END
```

Си

```
void main(void)
{ float x,y;
scanf("%f%f",&x,&y);
if (y<=x)
if (y<=-x)
if (y>=x*x-2)
  printf("принадлежит");
else
  printf("не принадлежит");
}
```

Последовательно выполните следующее:

1. Приведите пример таких чисел x, y , при которых программа неправильно решает поставленную задачу.

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой правильный способ доработки исходной программы.)

Решение.

Элементы ответа.

1. Пример: $x=2, y=2$. (Любая пара (x, y) , для которой выполняется: $y>x$ или $y>-x$.)
2. Возможная доработка (Паскаль, разбиение области на две части прямой $x=0$):


```
if (y>=x*x-2) and (y<=x) and (x>=0) or (x<=0) and (y<=-x) and (y>=x*x-2) then
  write('принадлежит')
else
  write('не принадлежит')
```

Возможная доработка (Си, разбиение на две пересекающиеся области):

```
if (y>=x*x-2 && (y<=x || y<=-x))
  printf("принадлежит");
else
  printf("не принадлежит");
```

Возможная доработка (Бейсик, отбрасывание части от большей области, используются вложенные условия):

```
IF y >= x * x - 2 THEN
IF NOT (y > x AND y > -x) THEN
PRINT "принадлежит"
ELSE
PRINT "не принадлежит"
ENDIF
ELSE
PRINT "не принадлежит"
ENDIF
```

Обратите внимание, что вариантов доработки может быть достаточно много, но обычно правильное описание заштрихованных областей в них представляет собой или объединение двух (или более) возможно пересекающихся областей, или исключение одной области из другой.

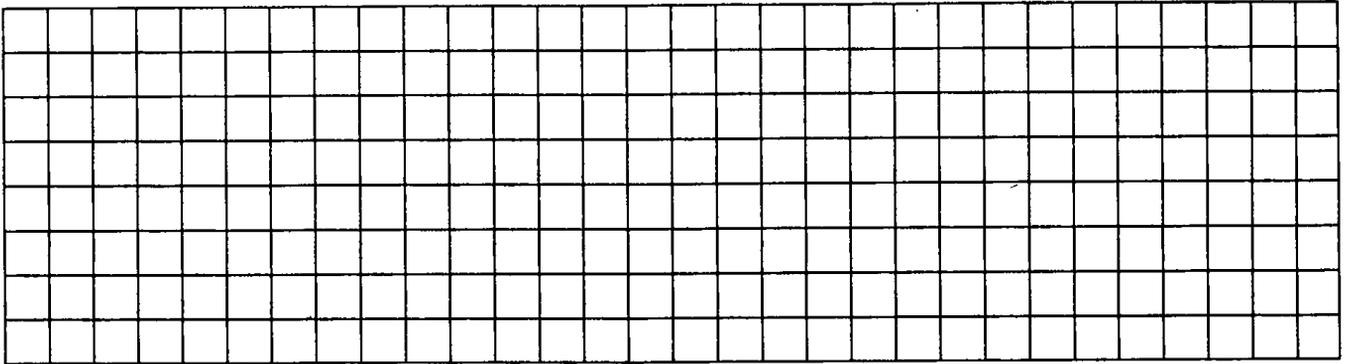
При разделении области вдоль какой-либо линии точки, которые лежат на этой линии внутри области, могут быть причислены к одной части, к другой или к обеим (т. е., например, в приведенном решении на языке Паскаль одно из условий $(x \geq 0)$ или $(x \leq 0)$ может быть строгим).

Решаем самостоятельно

C1-2009

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y — действительные числа) и определяет принадлежность точки заштрихованной области (рис. 14.2), включая ее границы. Программист торопился и написал программу неправильно.

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y — действительные числа) и определяет принадлежность точки заштрихованной области (рис. 4.1), включая ее границы. Программист торопился и написал программу неправильно.


C1-2010

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y — действительные числа) и определяет принадлежность точки заштрихованной области (рис. 14.3), включая ее границы. Программист торопился и написал программу неправильно.

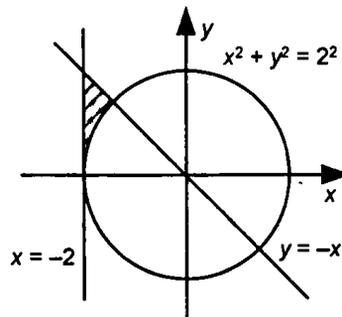


Рис. 14.3. Рисунок к заданию C1

Паскаль

```
var x, y: real;
begin
  readln(x, y);
  if x*x + y*y >= 4 then
    if x >= -2 then
      if y <= -x then
        write('принадлежит')
      else
        write('не принадлежит')
      end.
end.
```

Бейсик

```
INPUT x, y
IF x*x + y*y >= 4 THEN
  IF x >= -2 THEN
    IF y <= -x THEN
      PRINT "принадлежит"
    ELSE
      PRINT "не принадлежит"
    ENDIF
  ENDIF
ENDIF
END
```

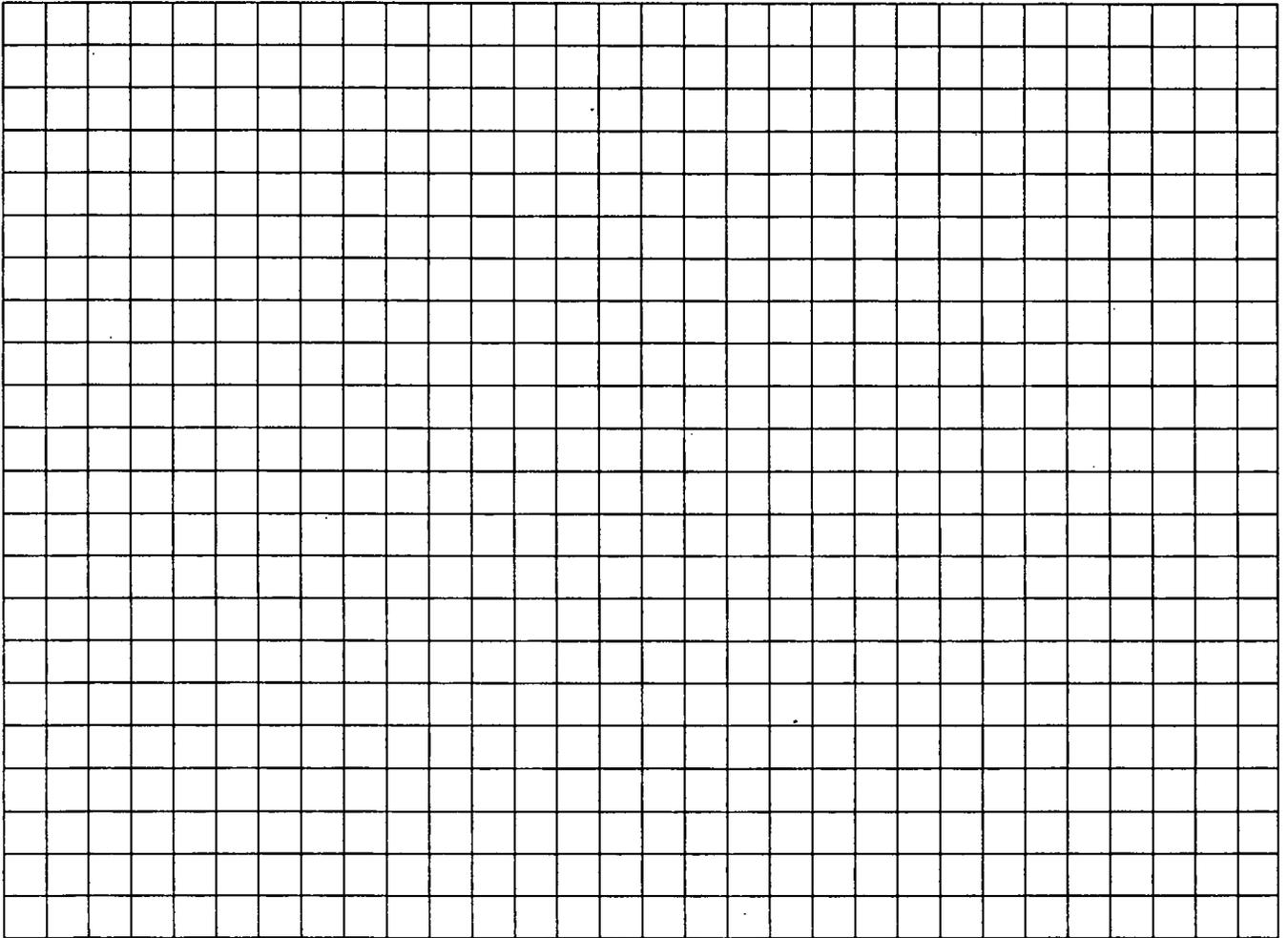
C++

```
void main(void)
{ float x, y;
  scanf("% f % f", &x, &y);
  if (x*x + y*y >= 4)
    if (x >= -2)
      if (y <= -x)
        printf("принадлежит");
      else
        printf("не принадлежит");
  }
```

Последовательно выполните следующее:

1. Приведите пример таких чисел x, y , при которых 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.)

Решение.



Дополнительные задания

Задание 14.1

Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной заштрихованной области (включая границы) — рис. 14.4.

Программист торопился и написал программу неправильно.

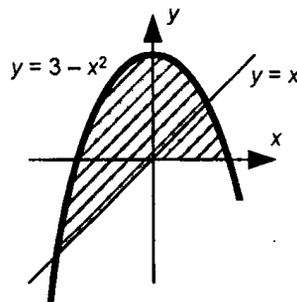


Рис. 14.4

Паскаль

```
var x, y: real;
begin
  readln(x,y);
  if y>=x then
    if y>=0 then
      if y<=3-x*x then
        write('принадлежит')
      else
        write('не принадлежит')
    end.
end.
```

Бейсик

```
INPUT x, y
IF y>=x THEN
  IF y>=0 THEN
    IF y<=3-x*x THEN
      PRINT "принадлежит"
    ELSE
      PRINT "не принадлежит"
    ENDIF
  ENDIF
ENDIF
END
```

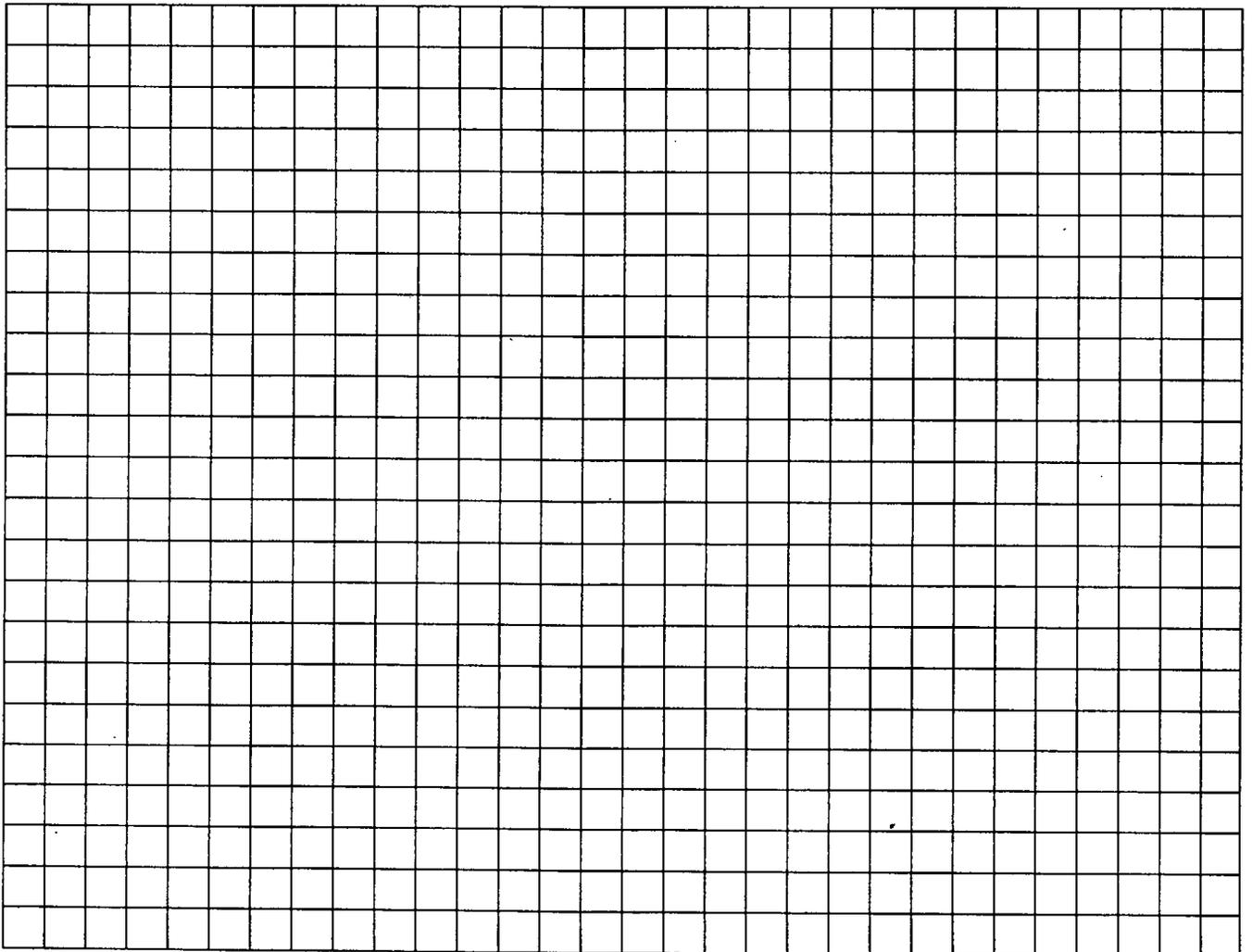
Си

```
void main(void)
{ float x,y;
  scanf("%f%f",&x,&y);
  if (y>=x)
    if (y>=0)
      if (y<=3-x*x)
        printf("принадлежит");
      else
        printf("не принадлежит");
  }
}
```

Последовательно выполните следующее:

1. Приведите пример таких чисел x , y , при которых программа неправильно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой правильный способ доработки исходной программы.)

Решение.



Задание 14.2

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y — действительные числа) и определяет принадлежность точки заштрихованной области (рис. 14.5), включая ее границы. Программист поторопился и написал программу неправильно.

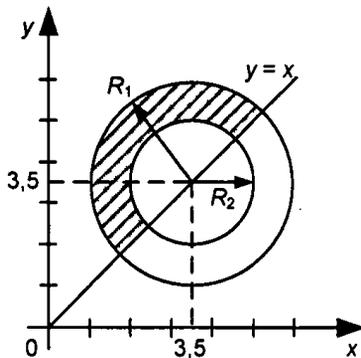


Рис. 14.5

Паскаль

```
var x, y, z: real;
begin
  readln(x,y);
  z := (x-3.5) * (x-3.5) + (y-3.5) * (y-3.5);
  if r2 * r2 <= z then
    if r1 * r1 >= z then
      if y >= x then
        write('принадлежит')
      else
        write('не принадлежит')
    end
  end
end.
```

Бейсик

```
INPUT x, y
z = (x-3.5) * (x-3.5) + (y-3.5) * (y-3.5)
IF r2^2 <= z THEN
  IF r1^2 >= z THEN
    IF y >= x THEN
      PRINT "принадлежит"
    ELSE
      PRINT "не принадлежит"
    ENDIF
  ENDIF
ENDIF
END
```

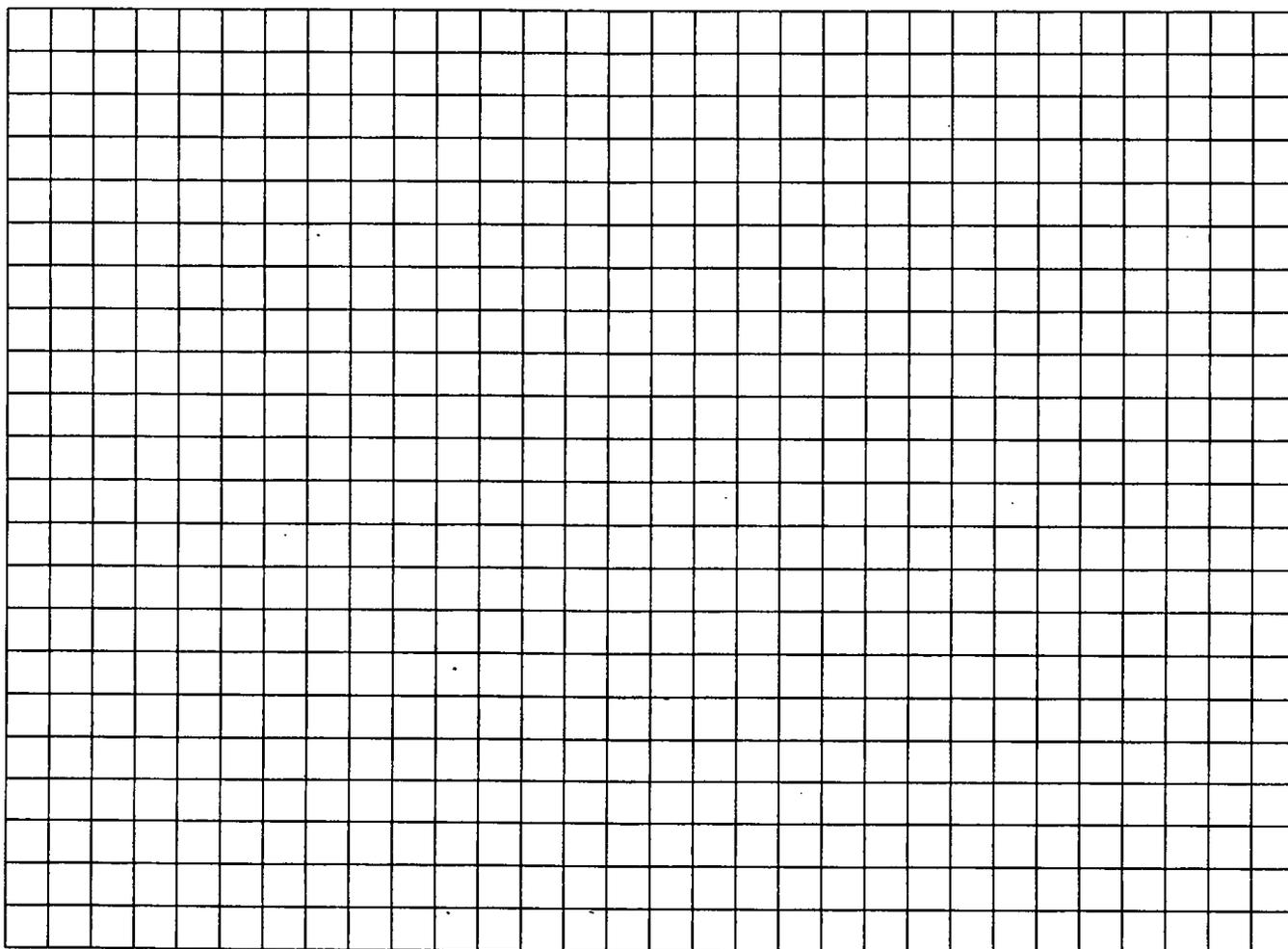
Си

```
void main(void)
{ float x, y, z;
  scanf ("%f%f", &x, &y);
  z = (x-3.5) * (x-3.5) + (y-3.5) * (y-3.5)
  if (r2 * r2 <= z)
    if (r1 * r1 >= z)
      if (y >= x)
        printf ("принадлежит");
      else
        printf "не принадлежит";
}
```

Последовательно выполните следующее:

1. Приведите пример таких чисел x , y , при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

Решение.



ГЛАВА 15

Написание простых программ

ТРЕБОВАНИЯ

Учащиеся должны знать этапы решения задачи на компьютере, основные алгоритмические конструкции и уметь применять их для обработки массивов данных. Необходимо уметь находить: минимальный и максимальный элементы массива, корни квадратного уравнения, сумму и произведение элементов массива, среднее арифметическое элементов массива, а также владеть простейшими способами сортировки, оперировать двумерными массивами и строковыми переменными.

Примеры заданий с решениями

C2-2011

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет подсчитать и вывести среднее арифметическое элементов массива, имеющих нечетное значение. Гарантируется, что в исходном массиве хотя бы один элемент имеет нечетное значение.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль

```
const
N=30;
var
a: array [1..N] of integer;
i, x, y: integer;
s: real;
begin
  for i:=1 to N do readln(a[i]);
  ...
end.
```

Бейсик

```
N=30
DIM A(N) AS INTEGER
DIM I, X, Y AS INTEGER
DIM S AS SINGLE
FOR I = 1 TO N
INPUT A(I)
NEXT I
...
END
```

C++

```
#include <stdio.h>
#define N 30
void main(void)
{int a[N];
int i, x, y;
float s;
```

Естественный язык

Объявляем массив А из 30 элементов.
Объявляем целочисленные переменные I, X, Y.
Объявляем вещественную переменную S.
В цикле от 1 до 30 вводим элементы массива А с 1-го по 30-й.

```

for (i=0; i<N; i++)          ...
    scanf("%d", &a[i]);
...
}

```

В качестве ответа 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 переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учетом синтаксиса и особенностей используемого вами языка программирования.

Решение.

Паскаль

```

x:=0;
y:=0;
for i:=1 to N do
if (a[i] mod 2=1) then begin
    x:=x+a[i];
    y:=y+1;
end;
s:=x/y;
writeln(s);

```

Бейсик

```

X = 0
Y = 0
FOR I = 1 TO N
IF A(I) MOD 2 = 1 THEN
X = X + A(I)
Y = Y + 1
ENDIF
NEXT I
S = X / Y
PRINT S

```

Си

```

x=0;
y=0;
for (i=0; i<N; i++)
if (a[i]%2==1)
{ x=x+a[i];
  y++;
}
s=(float)x/y;
printf("%f", s);

```

Естественный язык

Записываем в переменные X и Y начальное значение, равное нулю. В цикле от первого элемента до тридцатого находим остаток от деления элемента исходного массива на два. Если этот остаток равен единице, то увеличиваем счетчик суммы X на значение текущего элемента массива, а счетчик количества Y на 1. Переходим к следующему элементу.

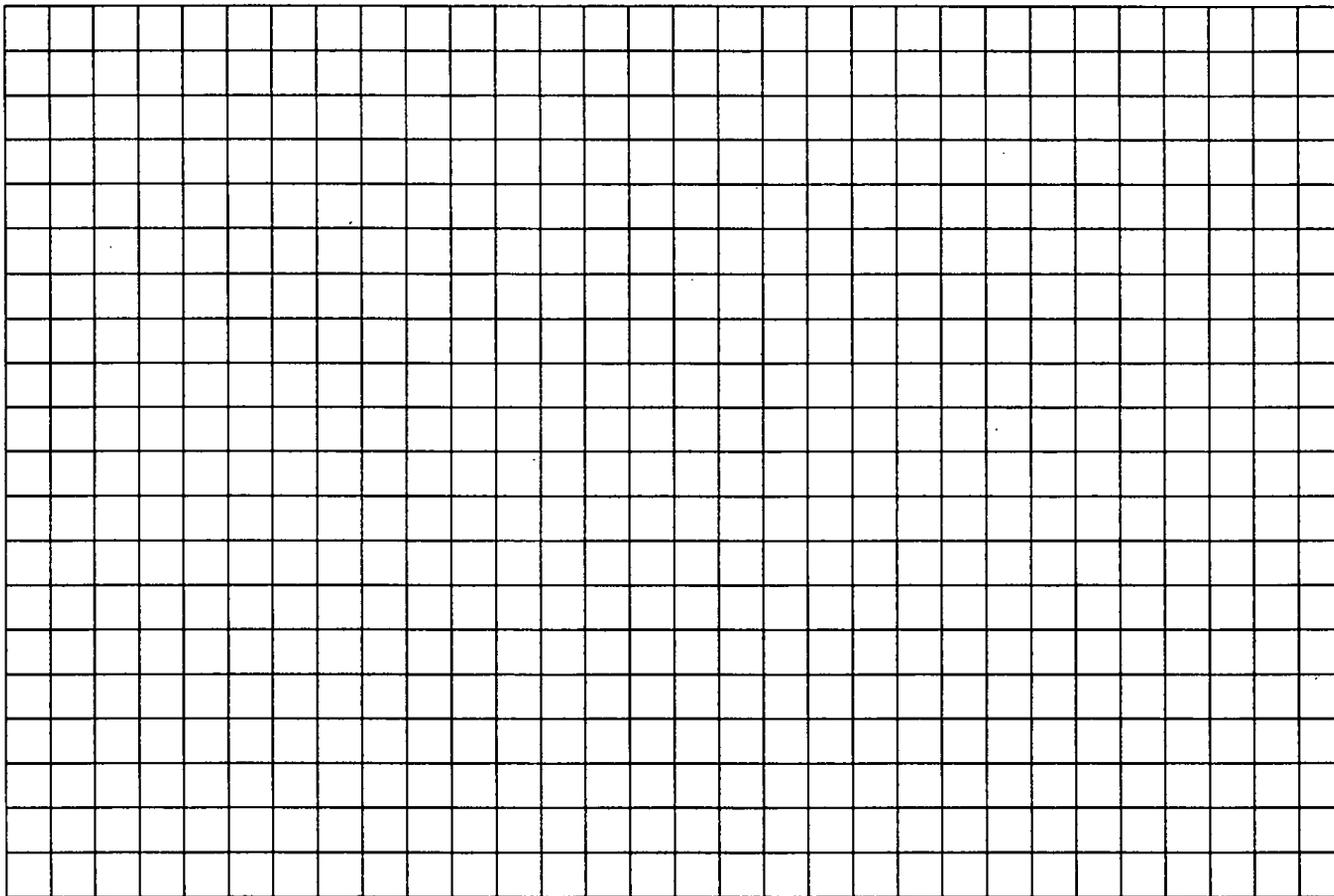
После цикла производим деление счетчика суммы X на счетчик количества Y и записываем результат в переменную S.

Выводим значение переменной S.

Решаем самостоятельно

C2-2009

Опишите на русском языке или на одном из языков программирования алгоритм получения из заданного целочисленного массива размером 30 элементов другого массива, который будет содержать модули значений элементов первого массива (не используя специальной функции, вычисляющей модуль числа).

Решение.**C2-2010**

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 находит и выводит минимальный балл среди учащихся, получивших за тест положительную оценку. Известно, что в классе хотя бы один учащийся получил за тест положительную оценку.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль

```
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.
```

Бейсик

```
N=30
DIM A(N) AS INTEGER
DIM I, J, MIN AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```


Дополнительные задания

Задание 15.1

Дан целочисленный массив из 100 элементов. Элементы могут принимать значения от -500 до 500 . Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет подсчитать и вывести среднее арифметическое элементов массива, оканчивающихся цифрой 7. Гарантируется, что в исходном массиве есть хотя бы один такой элемент.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль

```

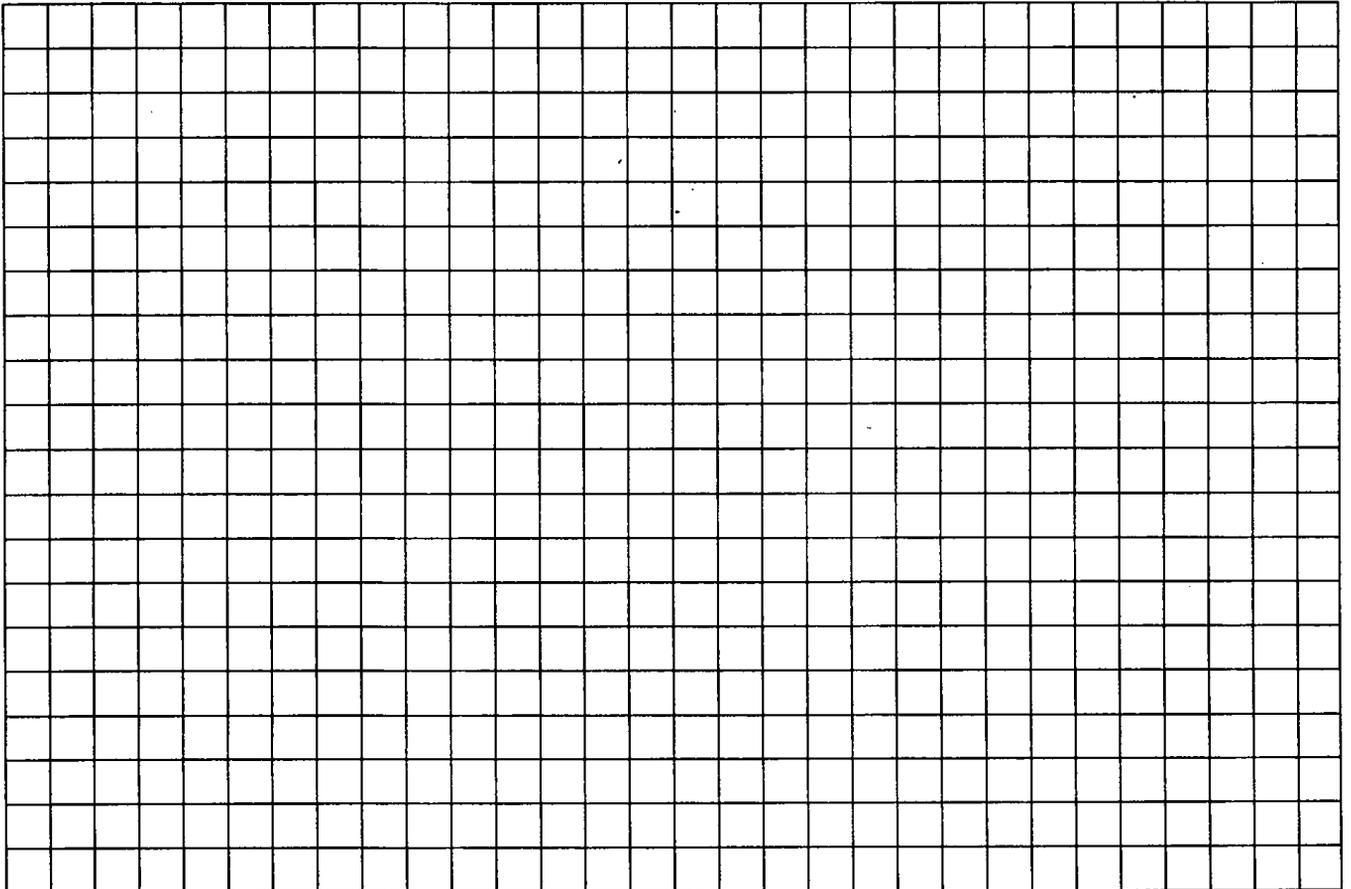
const
N=100;
var
a: array [1..N] of integer;
i, x, m: integer;
s: real;
begin
for i:=1 to N do readln(a[i]);
...
end.
```

Бейсик

```

N=100
DIM A(N) AS INTEGER
DIM I, X, M AS INTEGER
DIM S AS SINGLE
FOR I = 1 TO N
INPUT A(I)
NEXT I
...
END
```

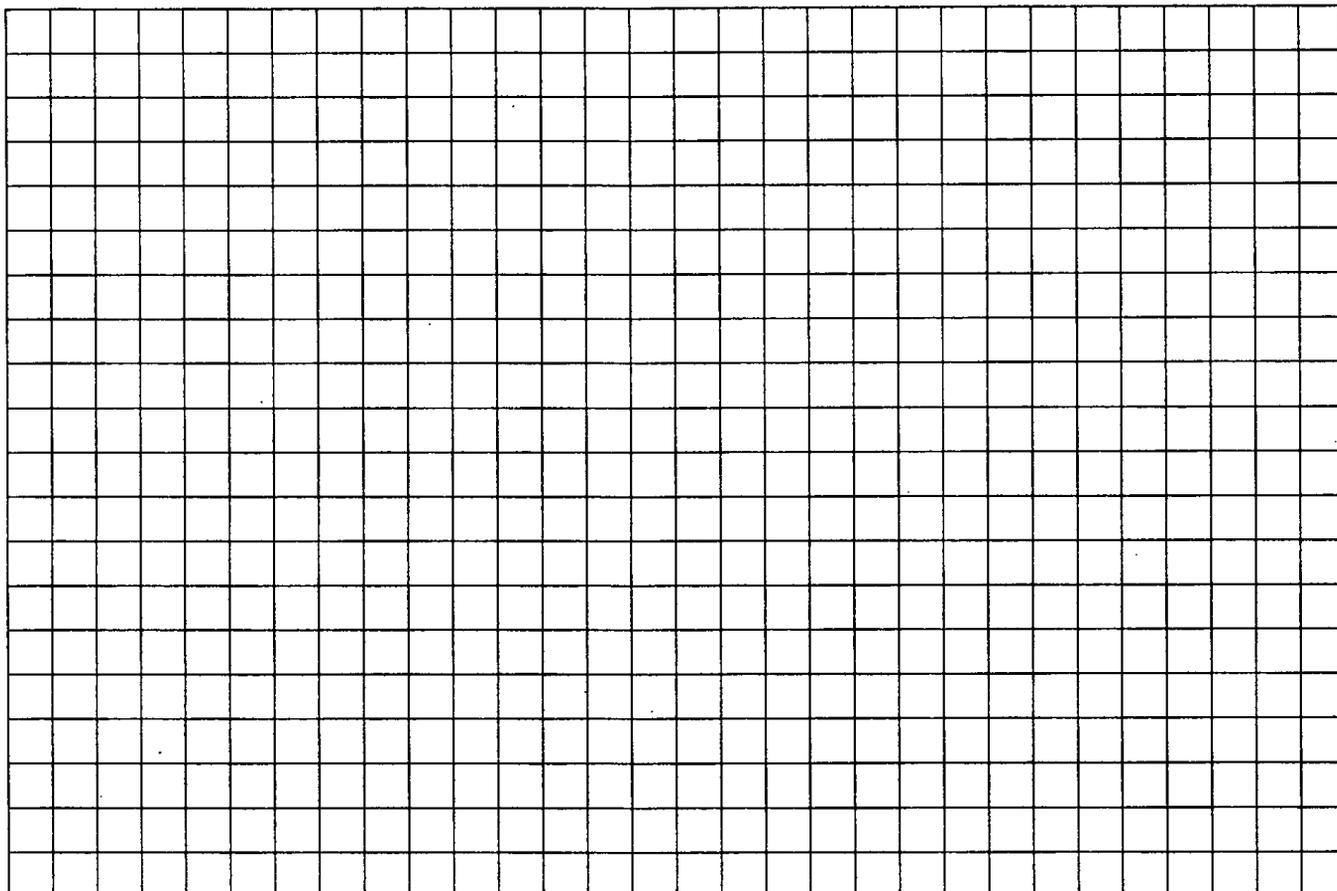
Решение.



Задание 15.2

Дан целочисленный массив из 100 элементов. Элементы могут принимать значения от 0 до 500. Опишите на русском языке или на одном из языков программирования алгоритм, который подсчитывает количество элементов массива, меньших среднего арифметического всех элементов массива.

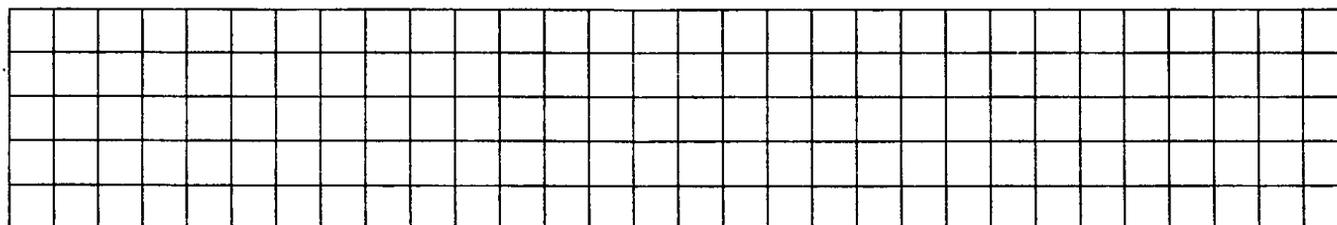
Решение.

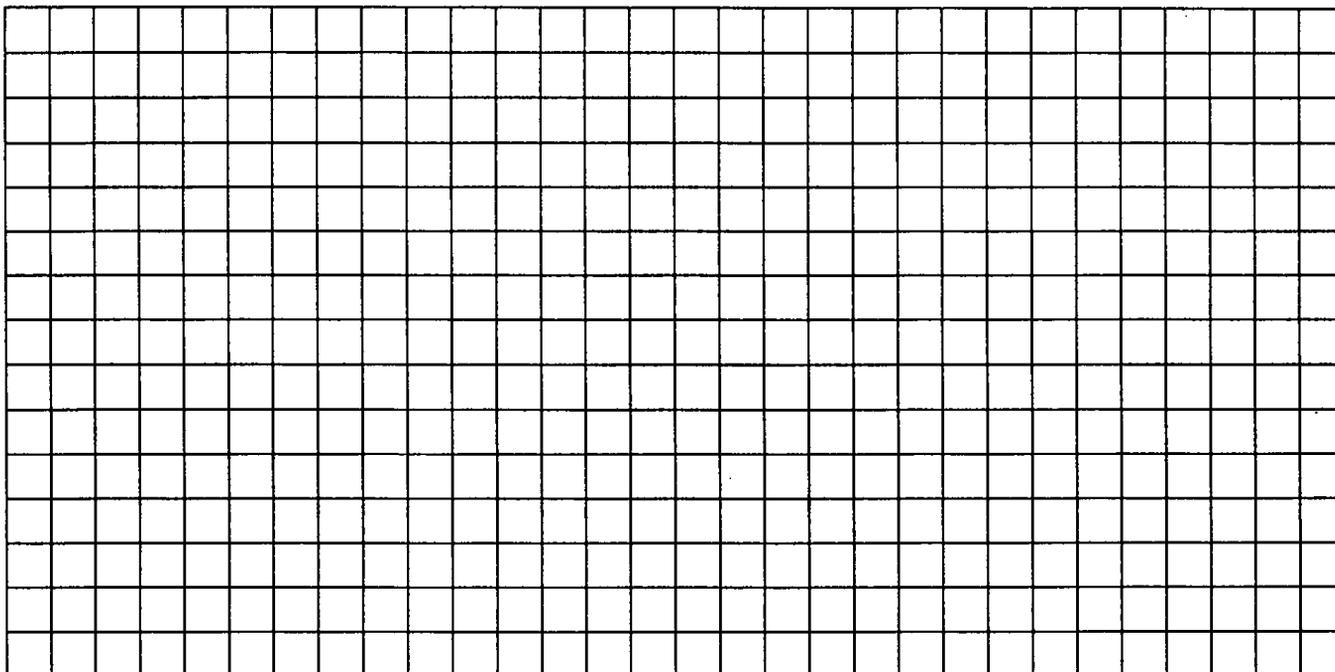


Задание 15.3

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 150 до 200 — рост учащихся выпускного класса. В баскетбольную команду входят все учащиеся, чей рост не менее 180 сантиметров. Гарантируется, что такие учащиеся в классе есть. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит рост самого низкого участника баскетбольной команды.

Решение.

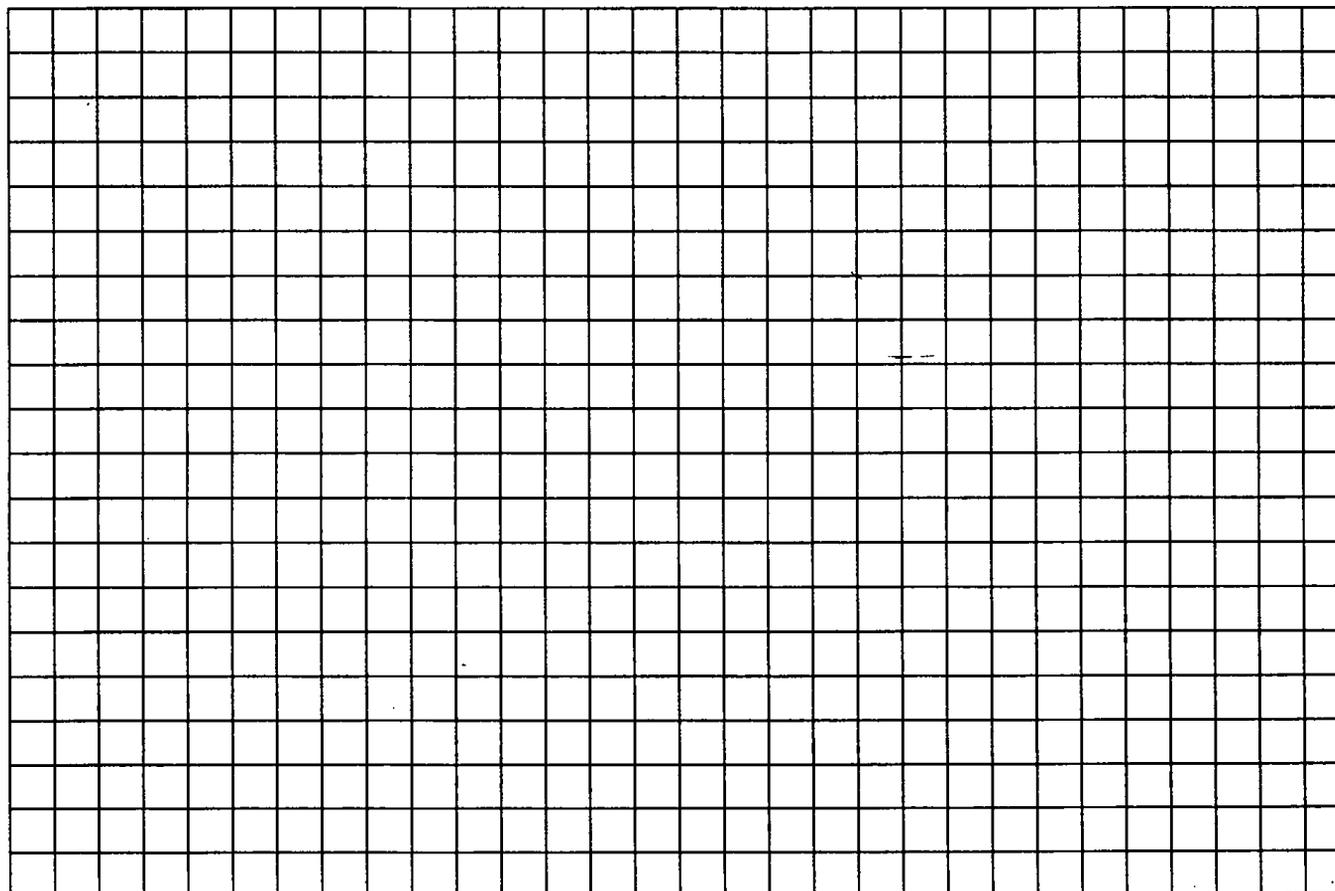




Задание 15.4

Опишите на русском языке или на одном из языков программирования алгоритм получения из заданного целочисленного массива размером 30 элементов другого массива, в котором поменяны местами все нечетные элементы с последующими за ними четными.

Решение.



ГЛАВА 16

Нахождение выигрышной стратегии

ТРЕБОВАНИЯ

Учащиеся должны уметь строить полное и неполное деревья логических игр с нахождением выигрышной стратегии в зависимости от первого хода.

Примеры заданий с решениями

C3-2011

Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 3, а во второй 4 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или удваивает число камней в какой-то кучке, или добавляет 4 камня в какую-то кучку. Игрок, после хода которого общее число камней в двух кучках становится больше 25, *проигрывает*.

Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Решение.

Выигрывает второй игрок.

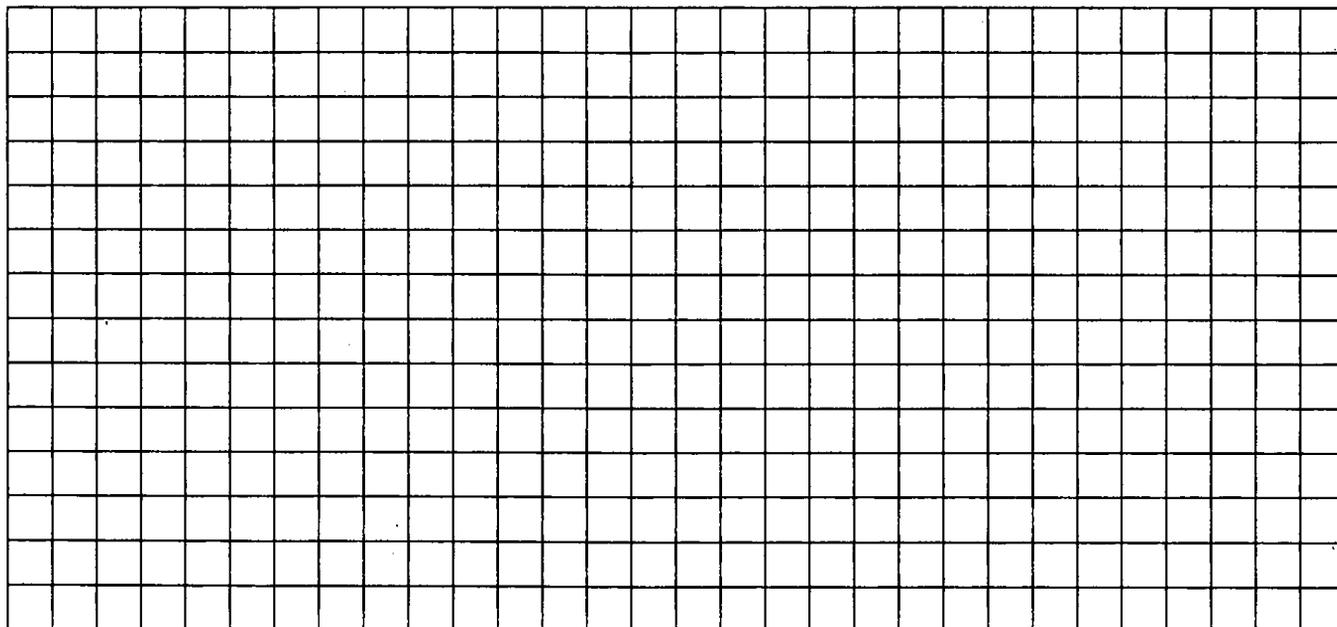
Для доказательства рассмотрим неполное дерево игры, оформленное в виде табл. 16.1, где в каждой ячейке записаны пары чисел, разделенные запятой.

Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучках соответственно.

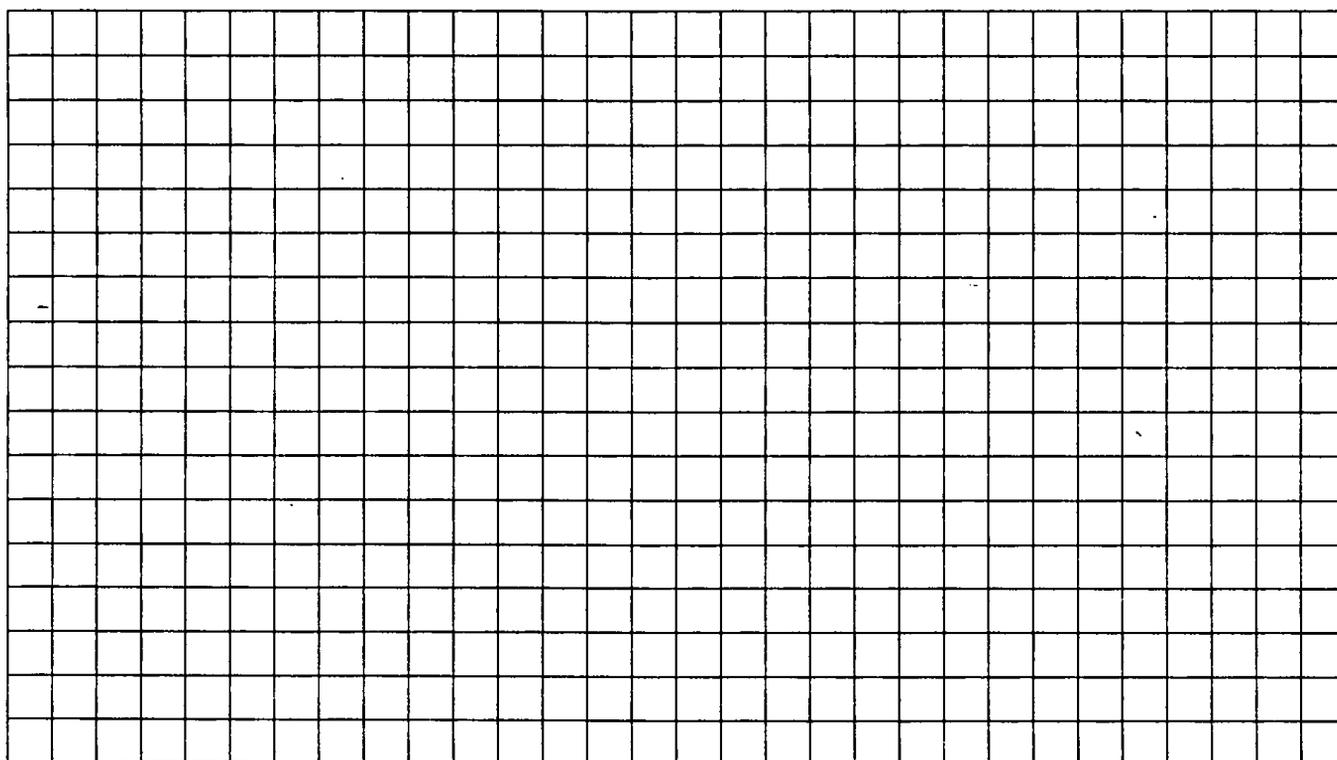
Решаем самостоятельно

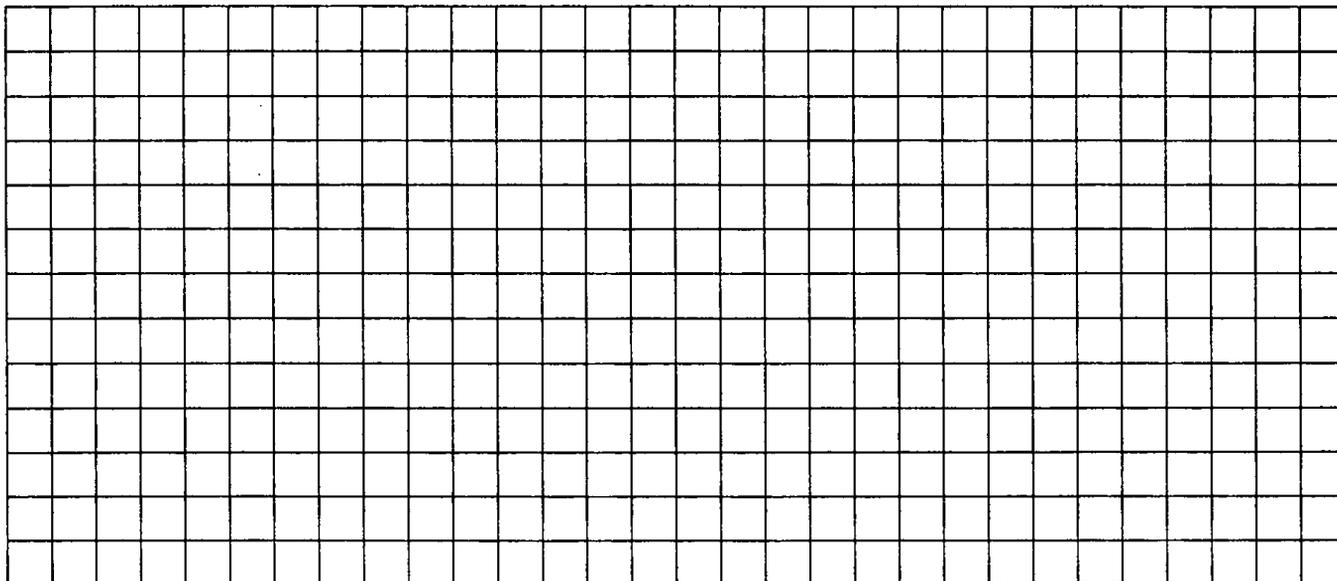
C3-2008

Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 1, а во второй 2 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или увеличивает в 3 раза число камней в какой-то кучке, или добавляет 2 камня в какую-то кучку. Выигрывает игрок, после хода которого общее число камней в двух кучках становится не менее 17 камней. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

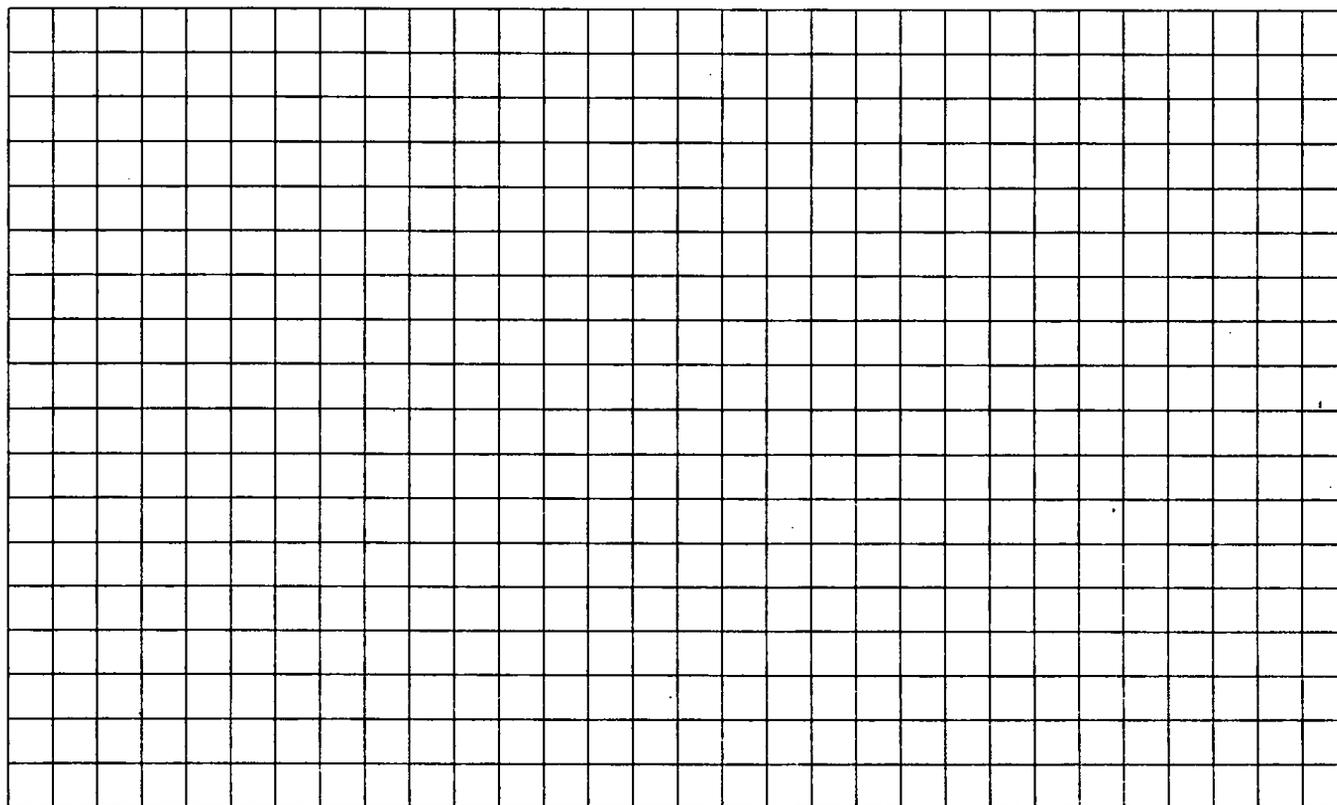
**C3-2009**

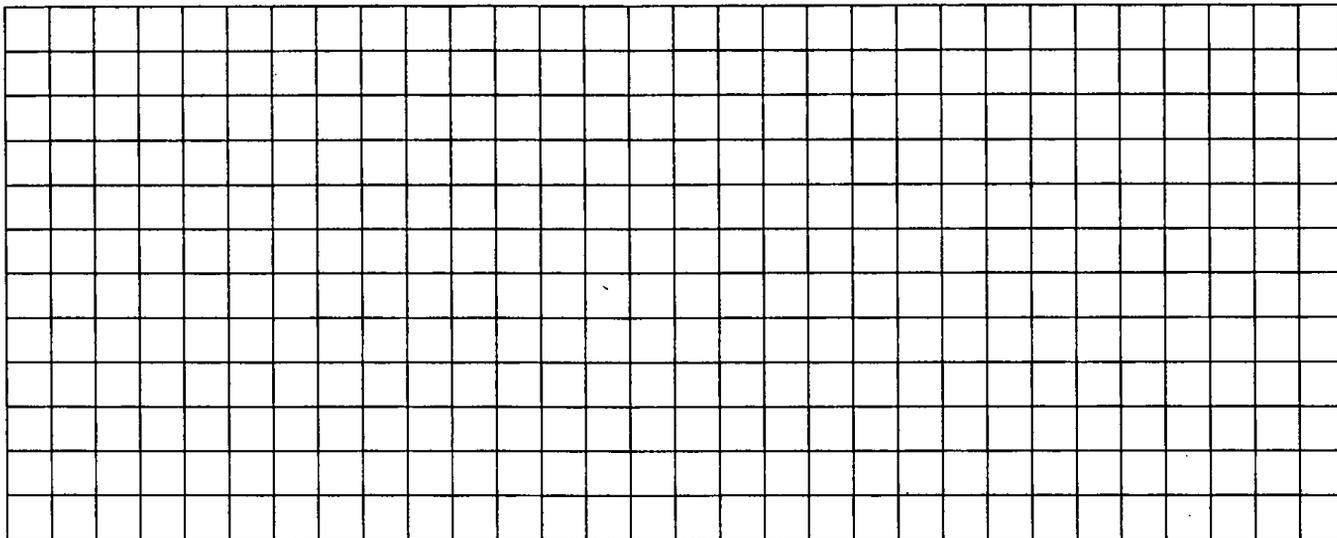
Два игрока играют в следующую игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. В начале игры фишка находится в точке с координатами $(5, 2)$. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: или в точку с координатами $(x + 3, y)$, или в точку с координатами $(x, y + 3)$, или в точку с координатами $(x, y + 4)$. Выигрывает игрок, после хода которого расстояние по прямой от фишки до точки с координатами $(0, 0)$ не меньше 13 единиц. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход?

Решение.

**С3-2010**

Два игрока играют в следующую игру. На координатной плоскости стоит фишка. В начале игры фишка находится в точке с координатами $(-2, -1)$. Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: $(x + 3, y)$, $(x, y + 4)$, $(x + 2, y + 2)$. Игра заканчивается, как только расстояние от фишки до начала координат превысит число 9. Выигрывает игрок, который сделал последний ход. Кто выигрывает при безошибочной игре — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Решение.

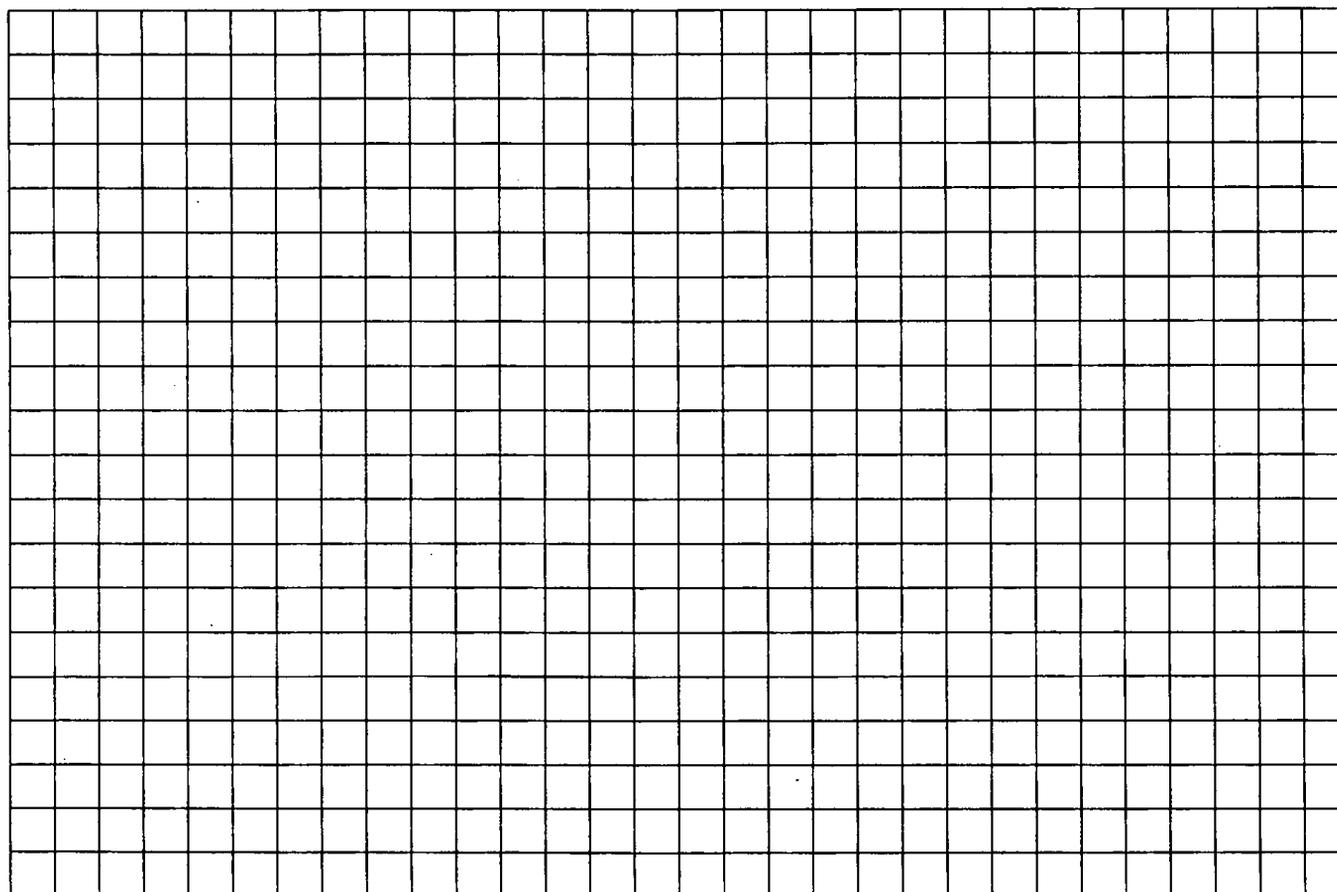


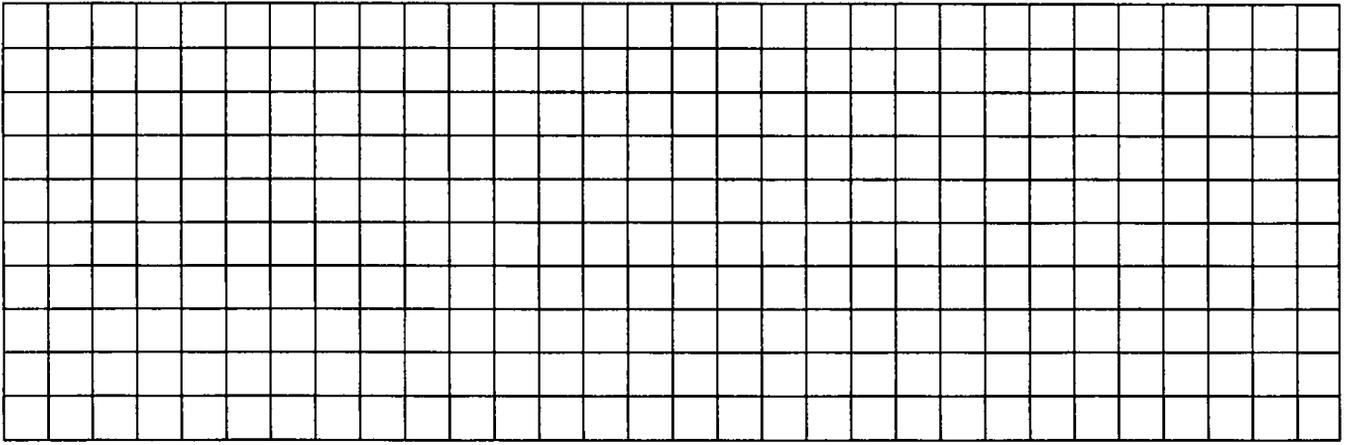
Дополнительные задания

Задание 16.1

Двое играют в следующую игру: имеются две кучки по 7 камней в каждой. За один ход разрешается взять любое количество камней из какой-либо одной кучки. Проигрывает тот, кто не может сделать ход. Кто выигрывает при правильной игре и как он должен играть? Ответ обоснуйте.

Решение.

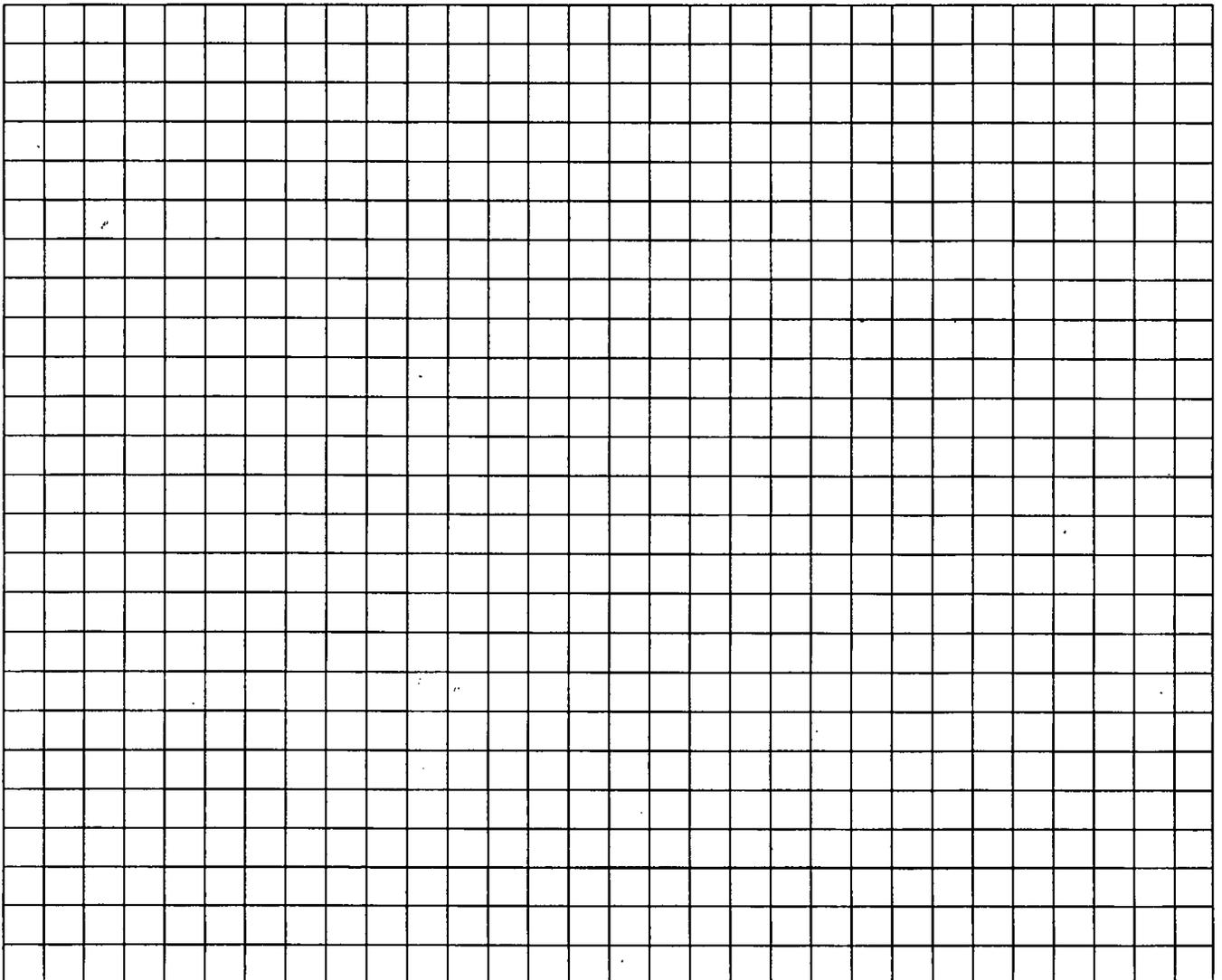


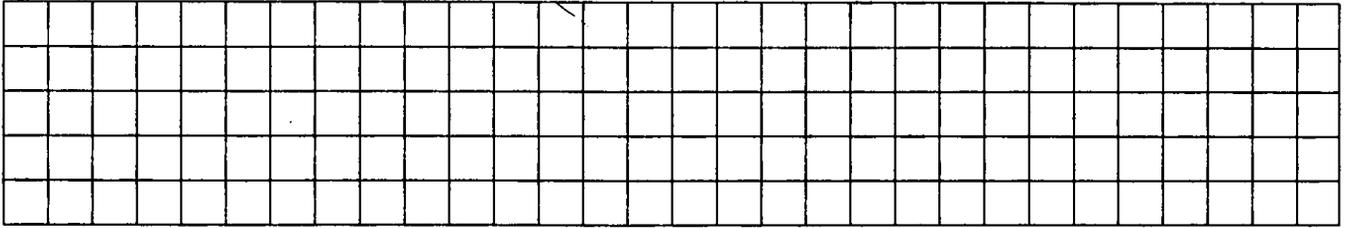


Задание 16.2

Числа от 1 до 20 выписаны в строчку. Два игрока по очереди расставляют между ними плюсы и минусы. После того как все места заполнены, подсчитывается результат. Если он четен, то выигрывает первый игрок, а если нечетен, то второй. Кто выиграет при правильной игре? Ответ обоснуйте.

Решение.

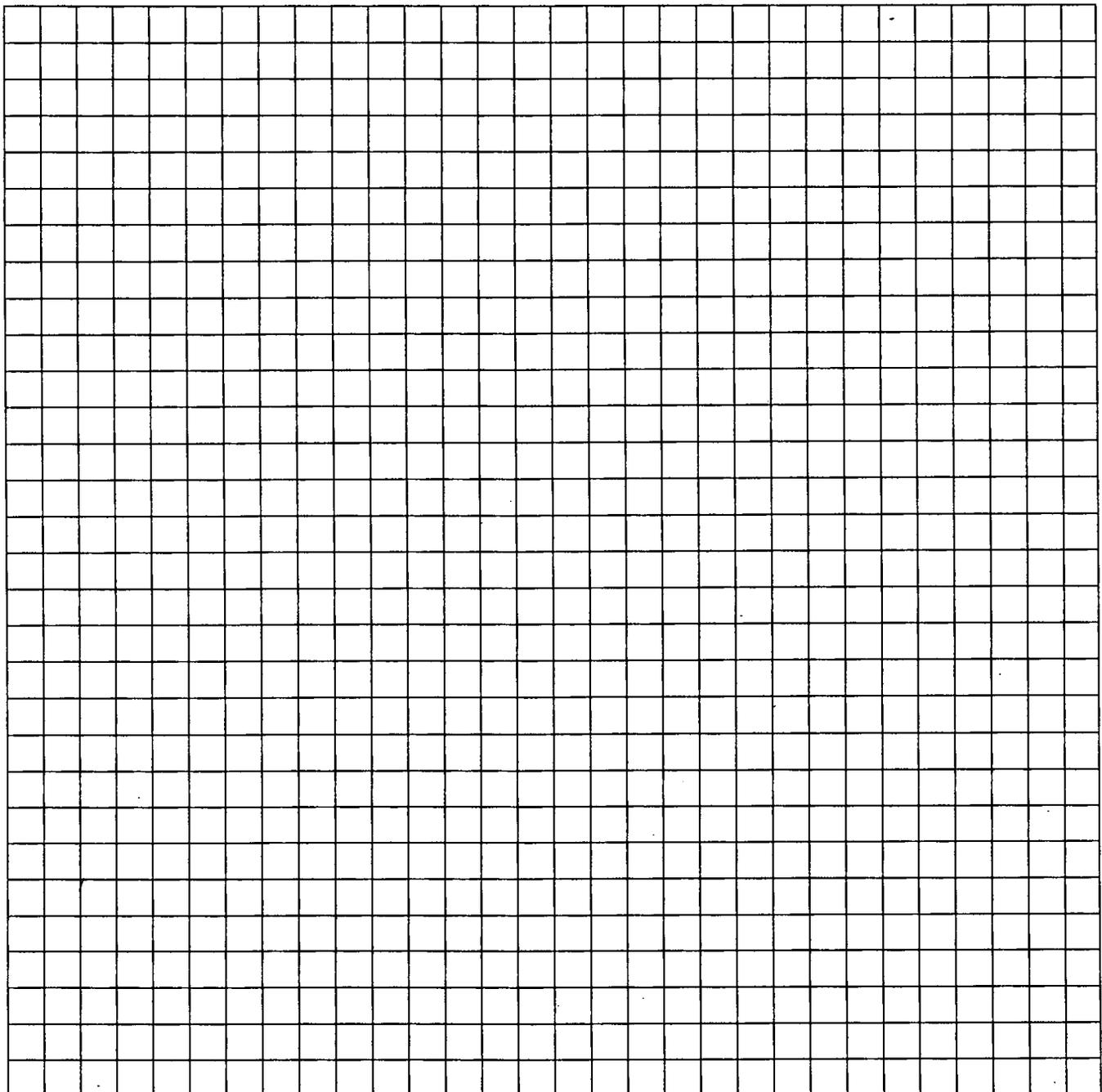




Задание 16.3

На столе лежат две кучки камней: в первой кучке 10 камней, а во второй — 15. За ход разрешается разделить любую кучку на две меньшие. Проигрывает тот, кто не сможет сделать ход. Может ли выиграть второй игрок? Ответ обоснуйте.

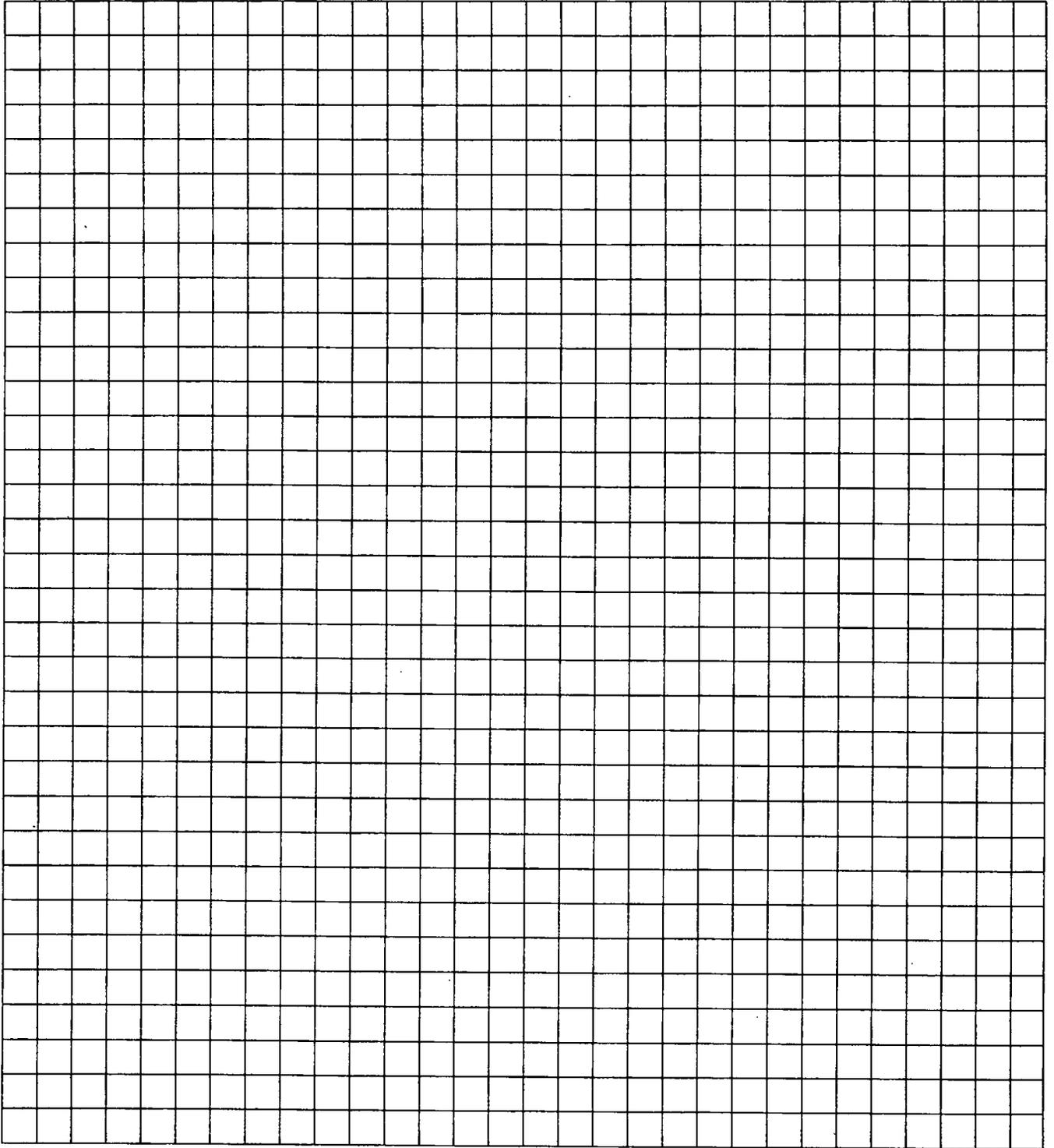
Решение.



Задание 16.4

Вспомним детство — игра "крестики-нолики" на поле 3×3 клетки. Кто выигрывает при правильной стратегии игры — "крестики" или "нолики"?

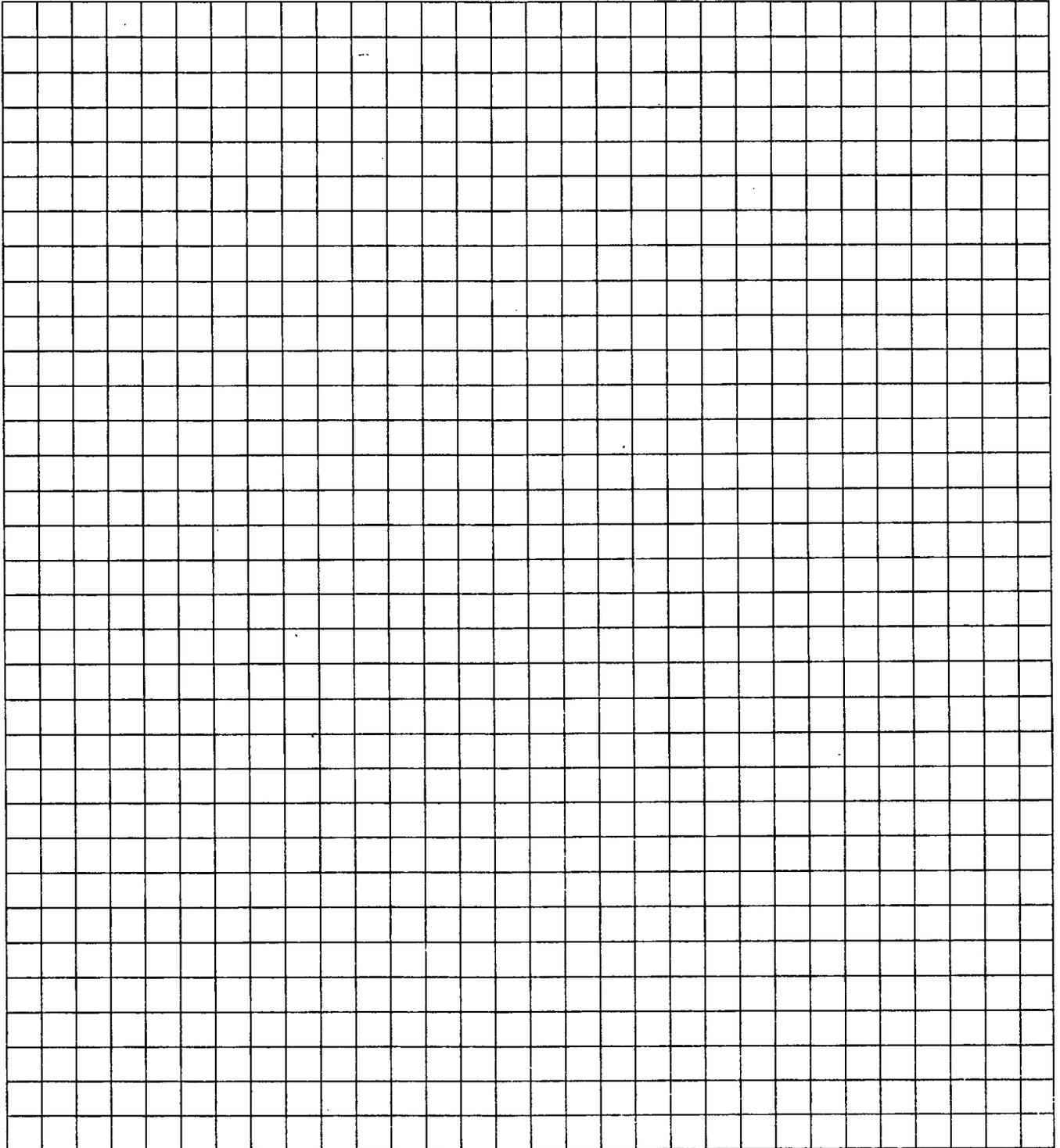
Решение.



Задание 16.5

В кучке 58 камней. Играют двое. За один ход разрешается взять из кучки любое количество камней от 1 до 4. Проигрывает тот, кто не может сделать ход. Кто выиграет при правильной игре?

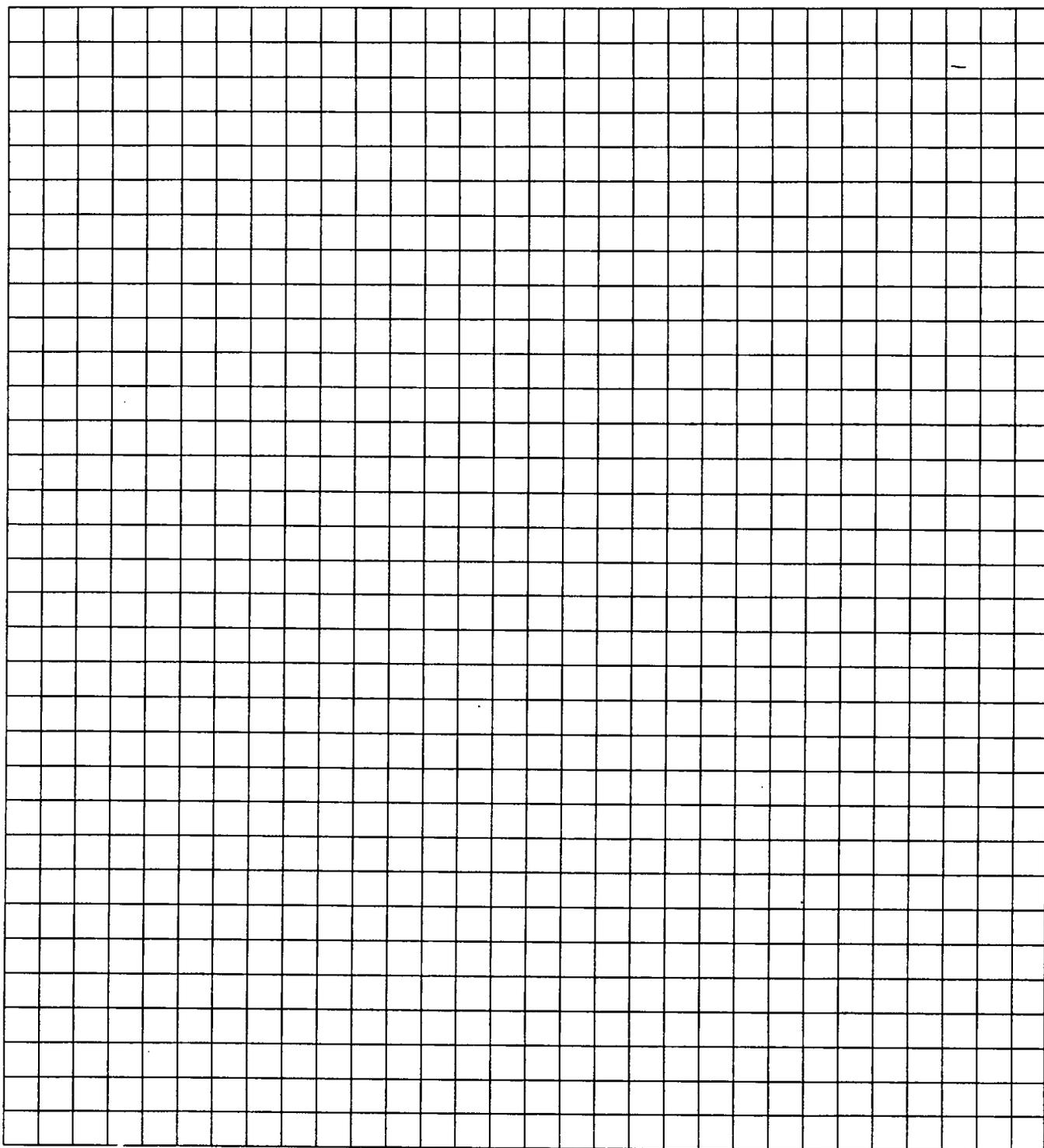
Решение.



Задание 16.6

Двое играют в следующую игру: имеются две кучки конфет. Играющие делают ход по очереди. Ход состоит в том, что играющий съедает одну из куч, а другую делит на две (равные или неравные) части. Если он не может разделить кучу, т. к. там всего одна конфета, то он ее съедает и выигрывает. Вначале в кучах было 33 и 35 конфет. Кто выигрывает: начинающий или его партнер, и как для этого надо играть?

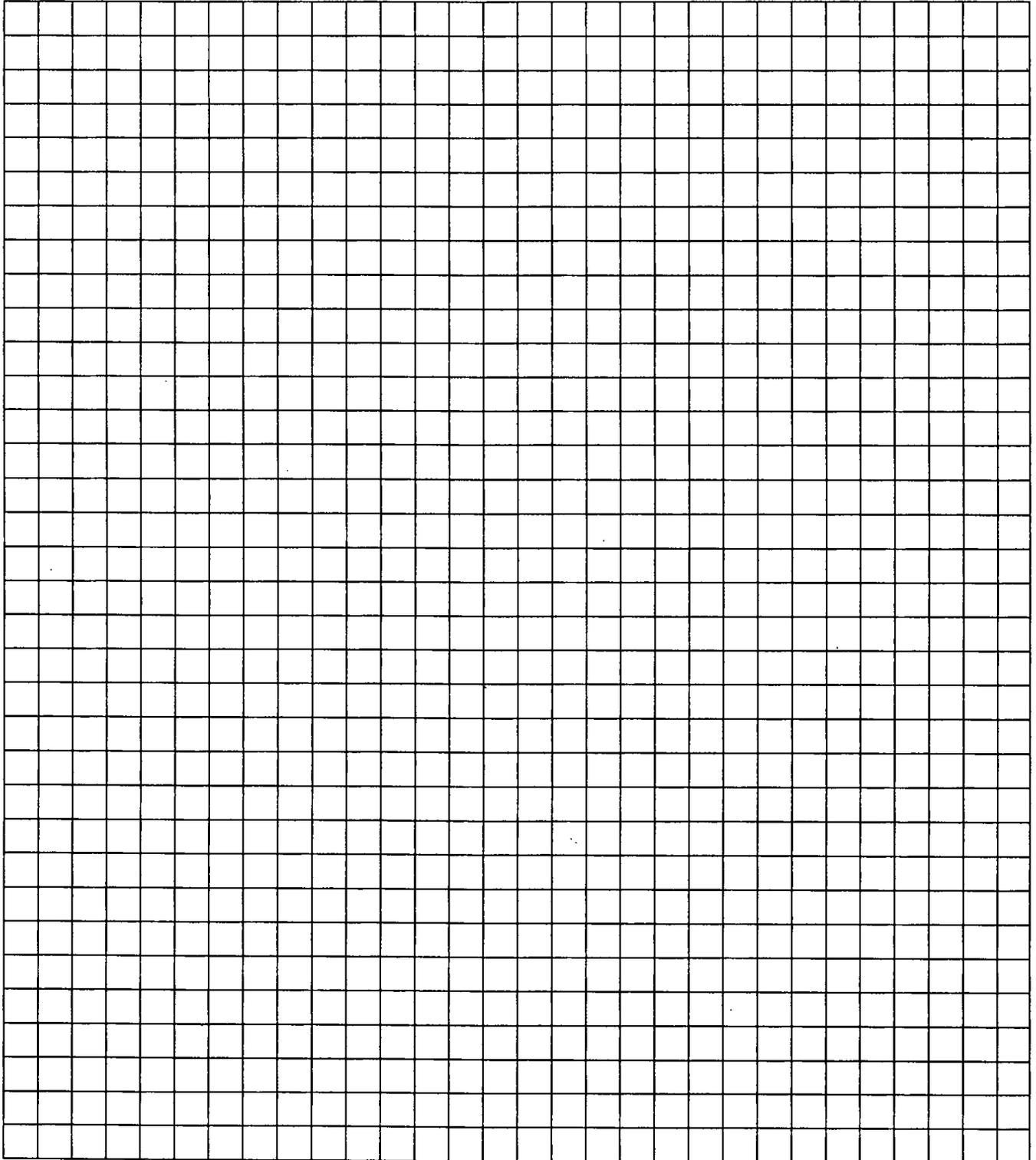
Решение.



Задание 16.7

Два кота украли цепочку из 6 сосисок и теперь делят ее. По очереди каждый кот перекусывает по одной перемычке между сосисками и съедает появившиеся при этом одиночные сосиски. Каждый хочет получить как можно больше. Кому сколько достанется?

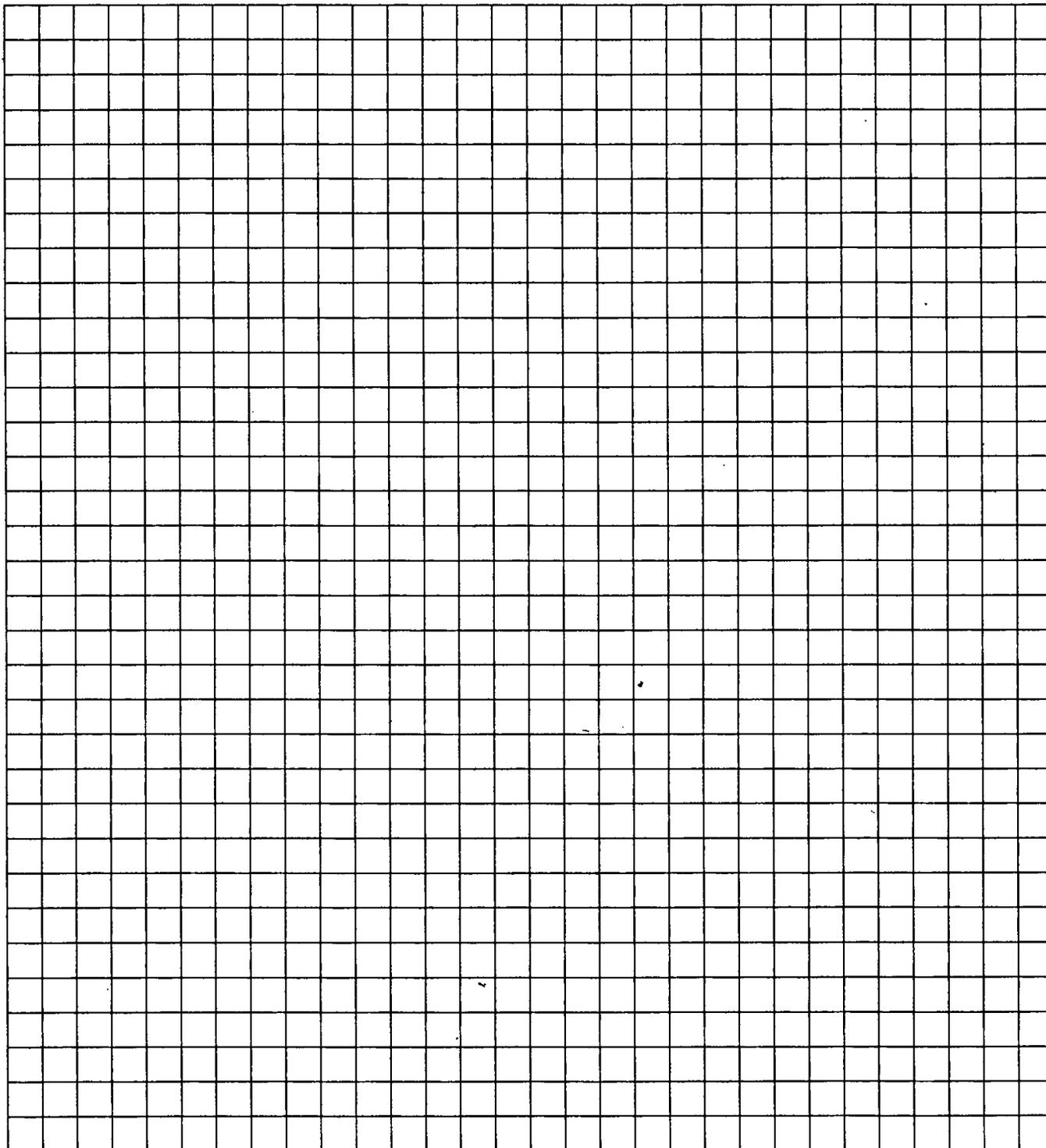
Решение.



Задание 16.8

Играют двое. В кучке 500 камней. За один ход разрешается взять количество камней, равное степени двойки (1, 2, 4, 8, ...). Выигрывает взявший последний камень. Кто выиграет при правильной игре и как он должен играть?

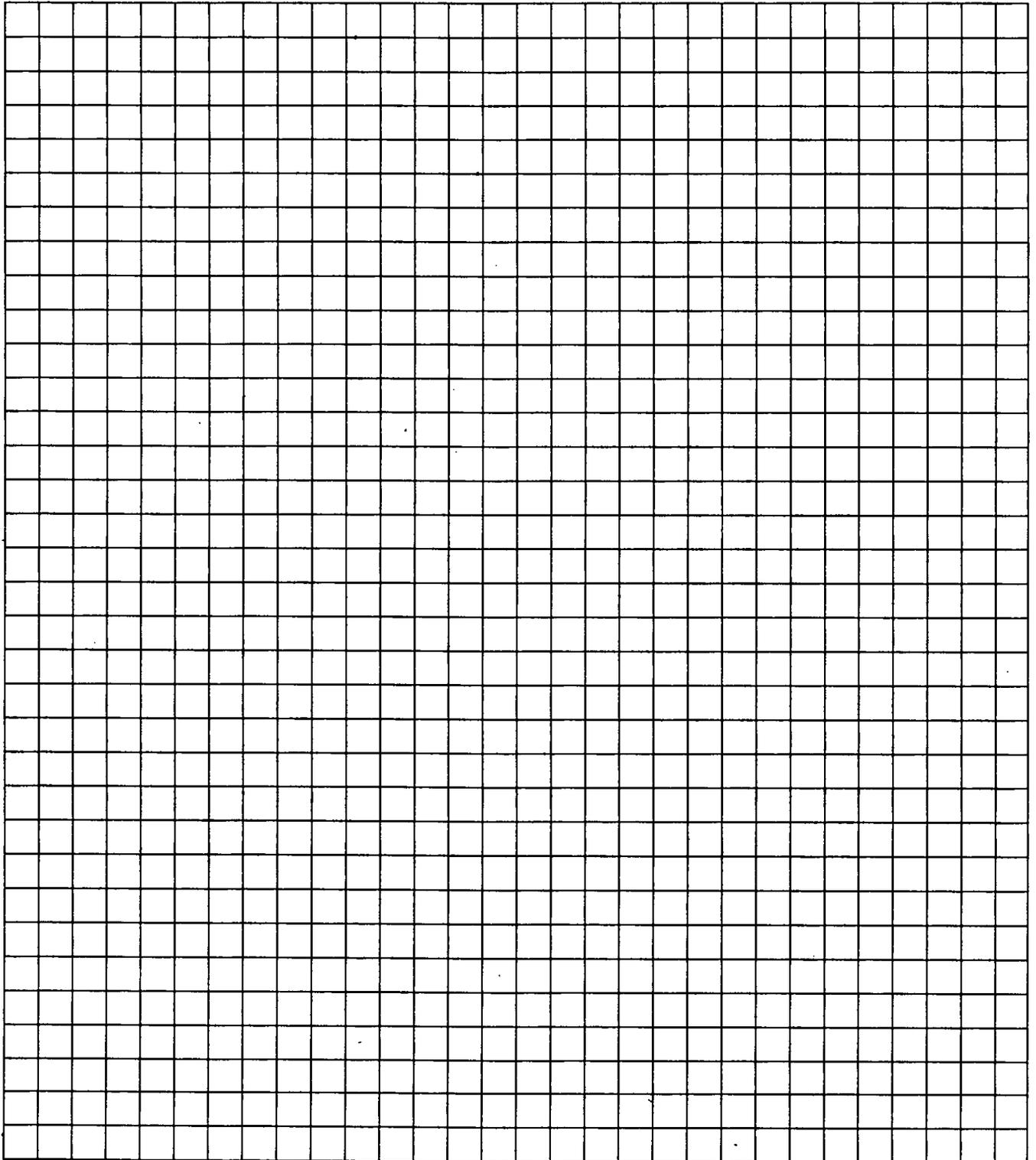
Решение.



Задание 16.9

Играют двое. У ромашки 11 лепестков. За один ход у нее отрывают один или два рядом растущих лепестка. Выигрывает тот, кто оторвет последний лепесток. Какой из игроков будет выигрывать при правильной стратегии?

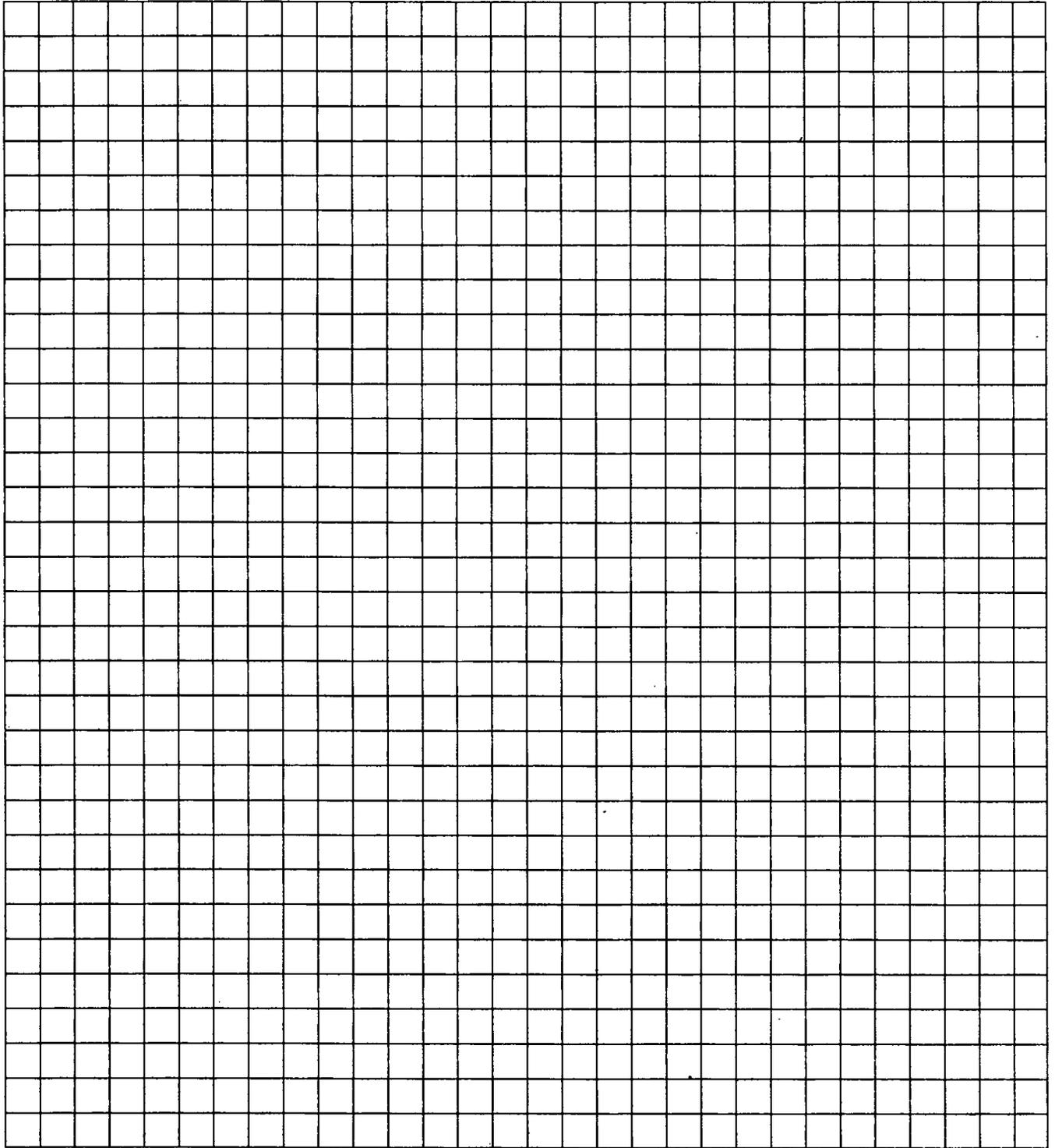
Решение.



Задание 16.10

В ряд лежат 25 монет. За ход разрешается брать одну или две рядом лежащие монеты. Проигрывает тот, кому нечего брать. Какой из игроков будет выигрывать при правильной стратегии?

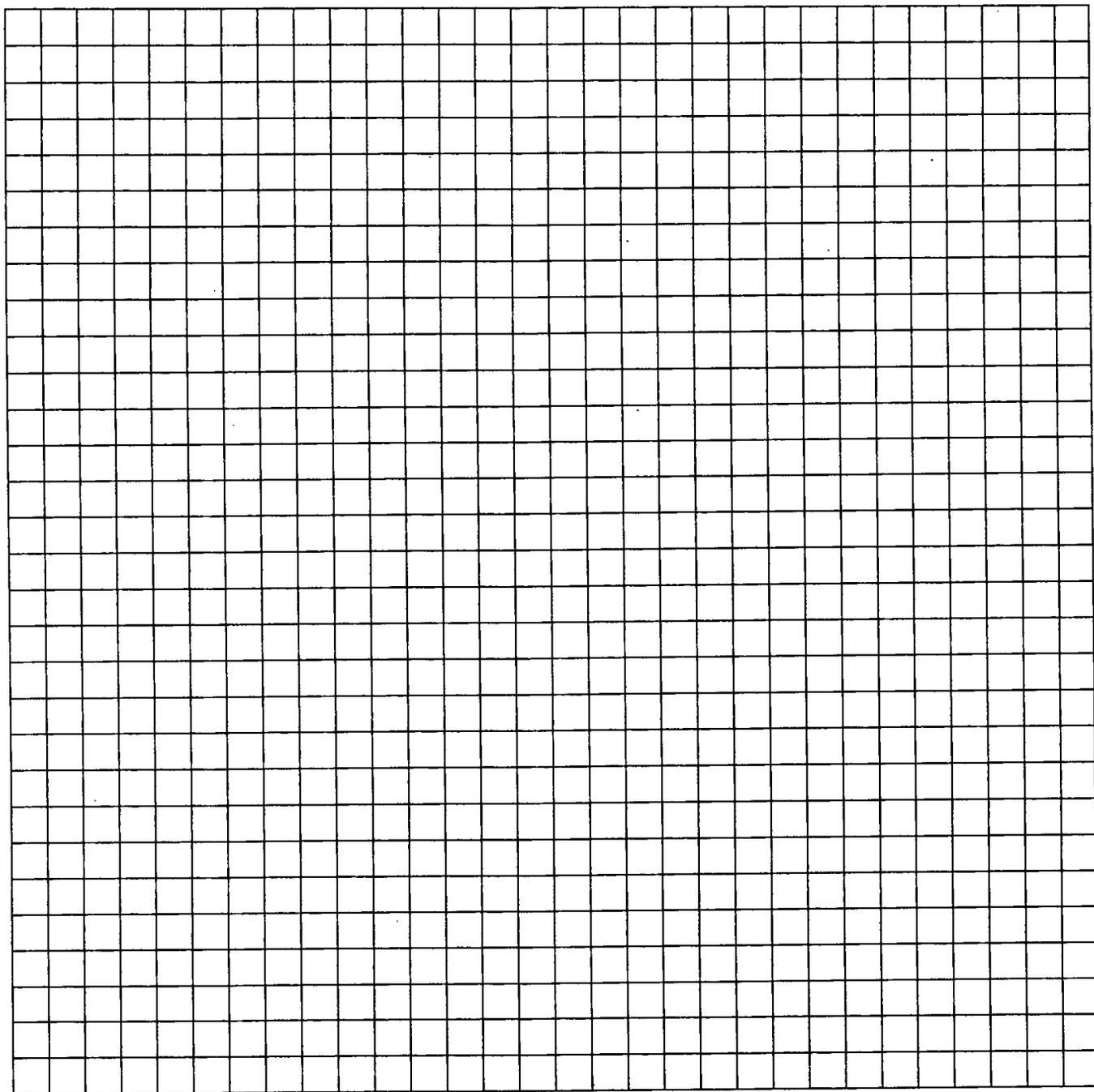
Решение.



Задание 16.11

Два игрока играют в следующую игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. В начале игры фишка находится в точке с координатами $(-3, 2)$. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: или в точку с координатами $(x + 5, y)$, или в точку с координатами $(x, y + 4)$, или в точку с координатами $(x + 3, y + 3)$. Выигрывает игрок, после хода которого расстояние по прямой от фишки до точки с координатами $(0, 0)$ больше 12 единиц. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Решение.



ГЛАВА 17

Написание сложных программ

Примеры заданий с решениями

C4-2011

На вход программе подается набор символов, заканчивающийся точкой (в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введена точка, или считывать данные из файла).

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая сначала будет определять, есть ли в этом наборе символы, соответствующие десятичным цифрам. Если такие символы есть, то можно ли переставить их так, чтобы полученное число было симметричным (читалось одинаково как слева направо, так и справа налево). Ведущих нулей в числе быть не должно, исключение — число 0, запись которого содержит ровно один ноль.

Если требуемое число составить невозможно, то программа должна вывести на экран слово "NO". А если возможно, то в первой строке следует вывести слово "YES", а во второй — искомое симметричное число. Если таких чисел несколько, то программа должна выводить максимальное из них. Например, пусть на вход подаются следующие символы:

```
Do not 911 to 09 do.
```

В данном случае программа должна вывести:

```
YES  
91019
```

Решение.

Пример правильной и эффективной программы на языке Паскаль:

```
var a:array['0'..'9'] of integer;  
    c, c_odd: char;  
    i, k: integer;  
    f: boolean;  
begin  
    for c:='0' to '9' do a[c]:=0;  
    read(c);  
    while c<>'.' do  
        begin  
            if c in ['0' .. '9'] then a[c] := a[c] + 1;  
        end  
        read(c);  
    end;
```

```
k := 0; {количество цифр, встречающихся нечетное число раз}
for c := '0' to '9' do
  if a[c] mod 2 = 1 then
    begin
      k := k + 1;
      c_odd := c
    end;
f := (a['0'] = 1);
for c := '1' to '9' do
  if (a[c] > 1) or (a[c] = 1) and (a['0'] = 0) then f := true;
  if (k > 1) or not f then writeln('NO') else
begin
  writeln('YES');
  for c := '9' downto '0' do
    for i := 1 to a[c] div 2 do
      write(c);
if k = 1 then
  write(c_odd);
for c := '0' to '9' do
  for i := 1 to a[c] div 2 do
    write(c);
end
end.
```

Решаем самостоятельно

C4-2009

На вход программе подаются сведения о номерах школ учащихся, участвовавших в олимпиаде. В первой строке сообщается количество учащихся N , каждая из следующих N строк имеет формат:

<Фамилия> <Инициалы> <номер школы>

где <Фамилия> — строка, состоящая не более чем из 20 символов, <Инициалы> — строка, состоящая из 4-х символов (буква, точка, буква, точка), <номер школы> — не более чем двузначный номер. <Фамилия> и <Инициалы>, а также <Инициалы> и <номер школы> разделены одним пробелом. Пример входной строки:

Иванов П.С. 57

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить на экран информацию, из какой школы было меньше всего участников (таких школ может быть несколько). При этом необходимо вывести информацию только по школам, пославшим хотя бы одного участника.

Следует учитывать, что $N \geq 1000$.

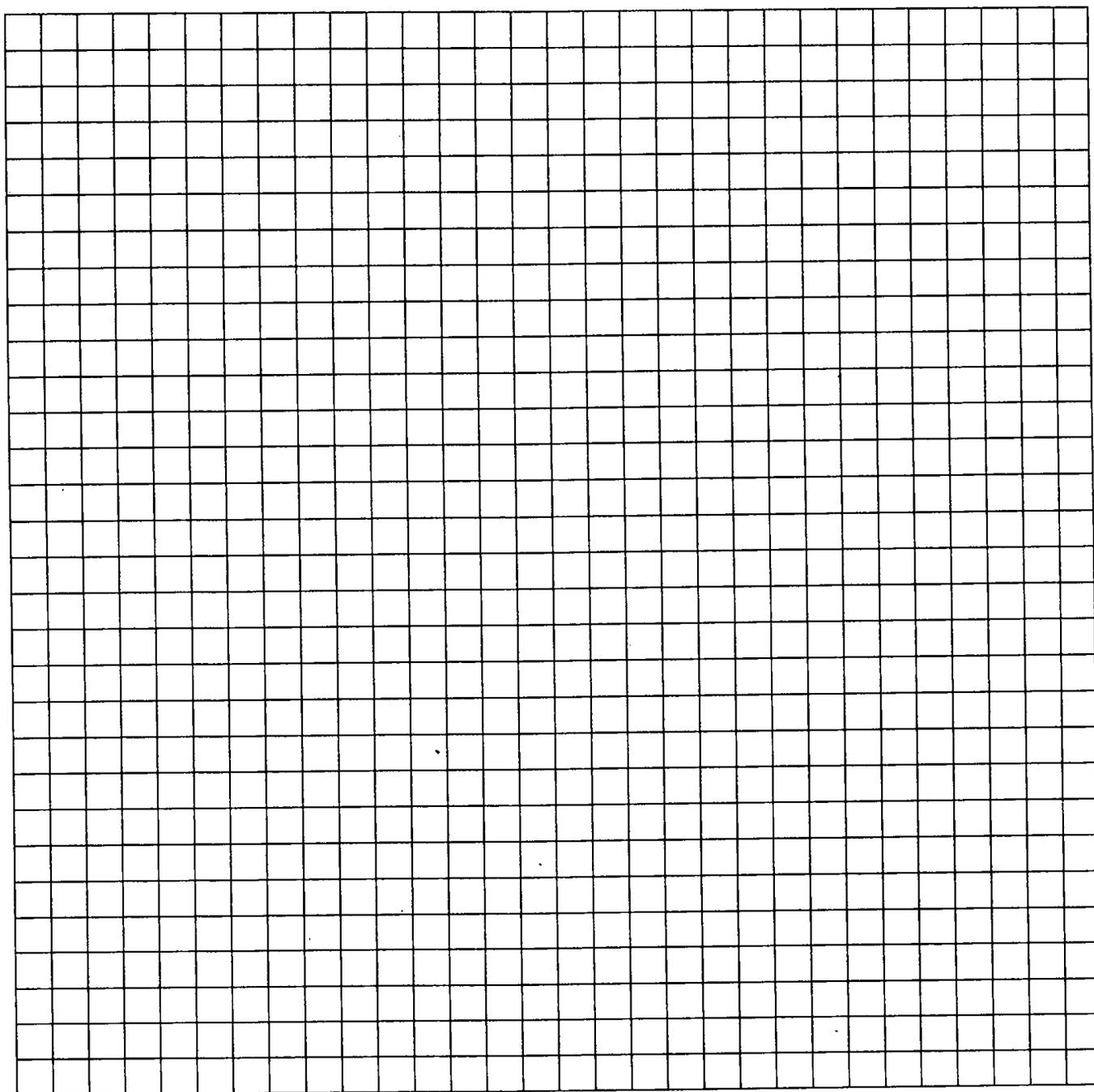
где <Компания> — строка, состоящая не более чем из 20 символов без пробелов, <Улица> — строка, состоящая не более чем из 20 символов без пробелов, <Марка> — одно из чисел — 92, 95 или 98, <Цена> — целое число в диапазоне от 1000 до 3000, обозначающее стоимость одного литра бензина в копейках. <Компания> и <Улица>, <Улица> и <Марка>, а также <Марка> и <Цена> разделены ровно одним пробелом. Пример входной строки:

Синойл Цветочная 95 2250

Программа должна выводить через пробел 3 числа: количество АЗС, продающих дешевле всего 92-й, 95-й и 98-й бензин соответственно. Если бензин какой-то марки нигде не продавался, то следует вывести 0. Пример выходных данных:

12 1 0

Решение.

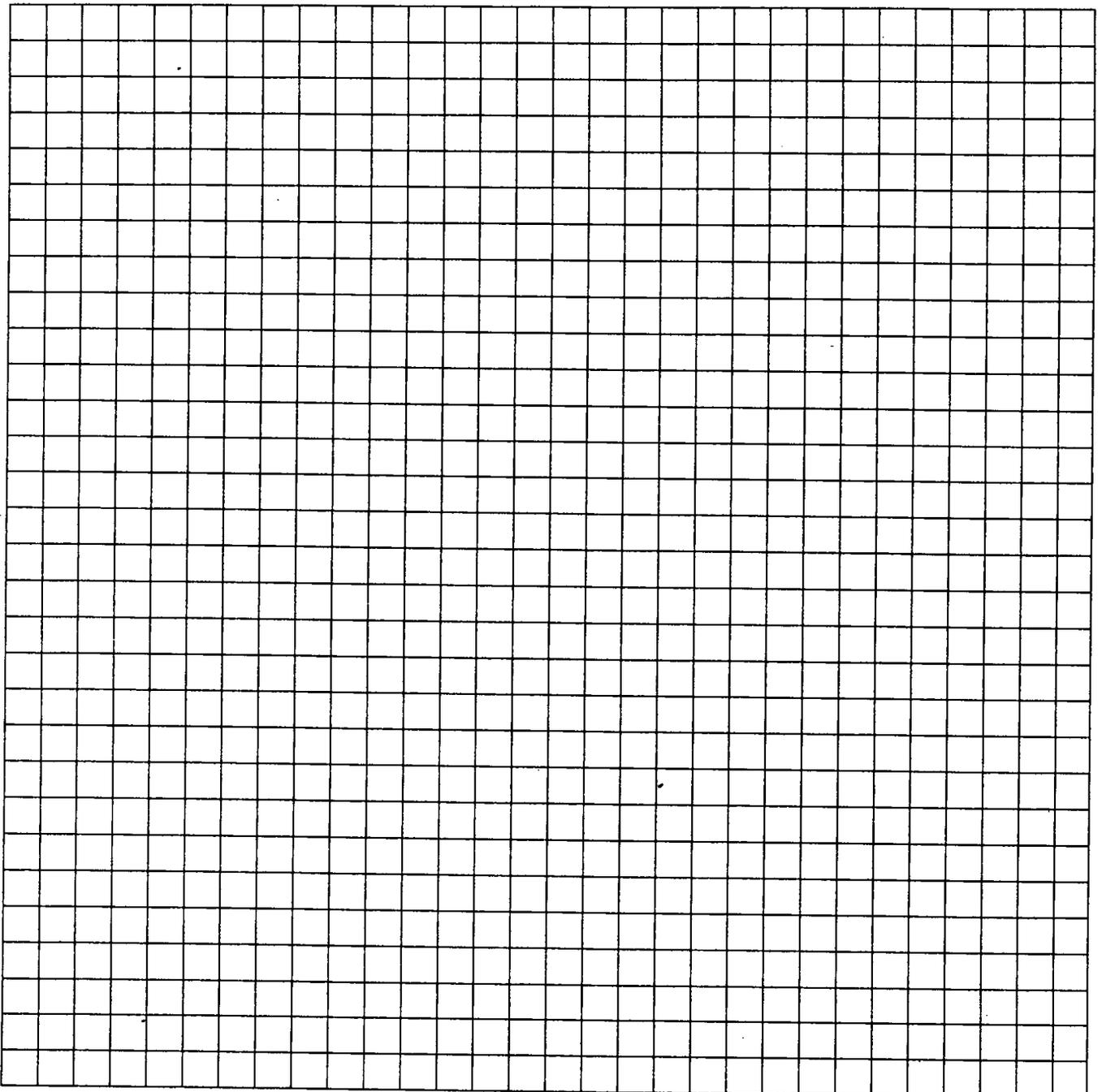


Дополнительные задания

Задание 17.1

Программа читает текст из файла один раз, сразу подсчитывая в массиве, хранящем 12 вещественных чисел, сумму температур в каждом из месяцев, одновременно суммируя все температуры в году. Затем с использованием этого массива ищется максимальное отклонение среднемесячной температуры от среднегодовой. За дополнительный просмотр этого массива распечатывается информация об искомым месяцах. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, месяц с максимальным отклонением единственен).

Решение.



Задание 17.2

Результаты выигрышей денежной лотереи представлены последовательностью натуральных чисел, записанных в текстовом файле в несколько строк через пробел. Три первые цифры каждого числа — номер билета, а последние три цифры — величина выигрыша.

Определите и выведите номера билетов с наибольшим выигрышем.

Например:

◆ входные данные:

10345859 1254386 132563

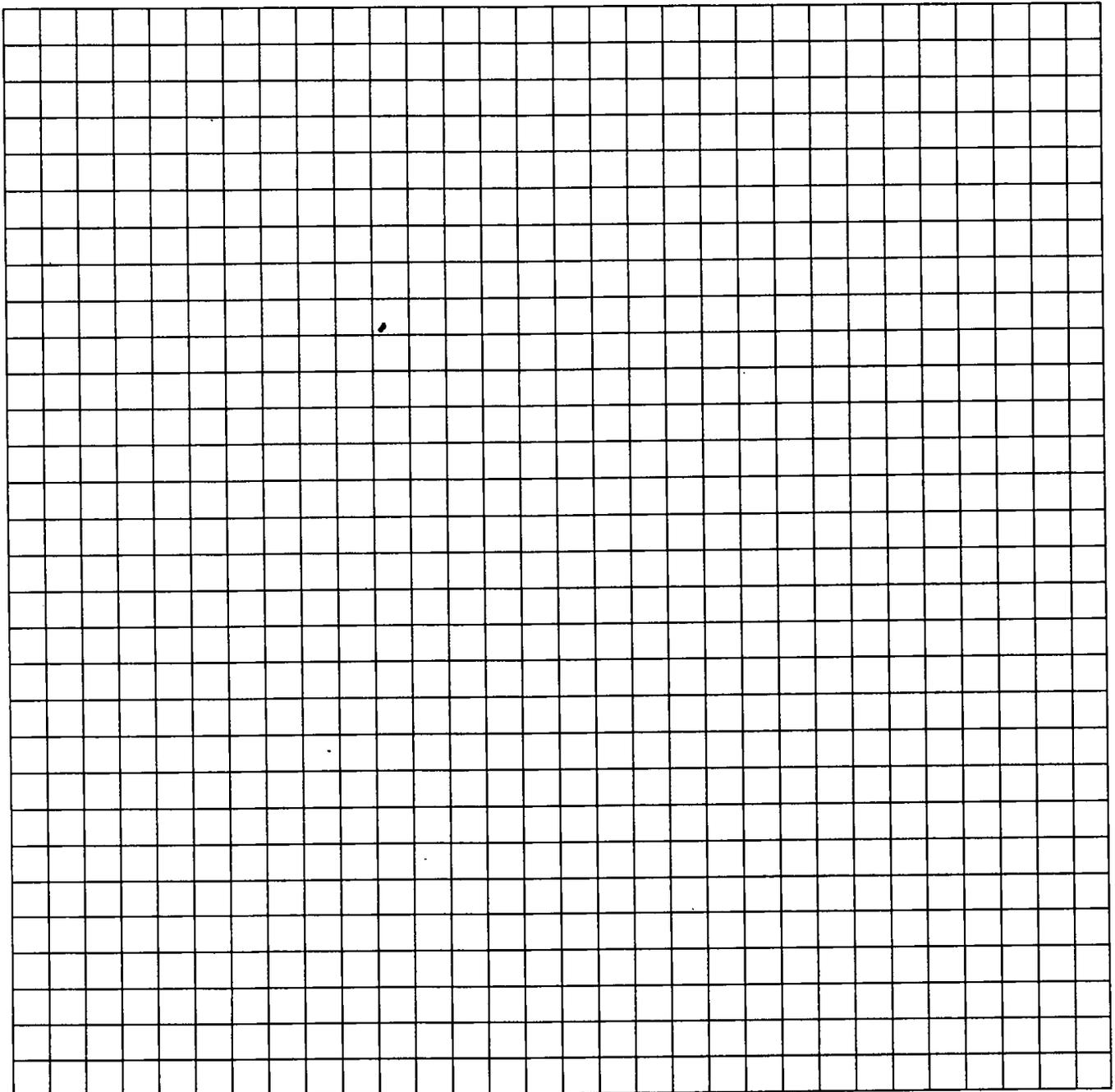
2377739 4237859

◆ выходные данные:

103 -859

423 -859

Решение.



ГЛАВА 18

Вариант ЕГЭ повышенной сложности

ЧАСТЬ А. Ответ, как выбор из представленных вариантов

A1

В кодировке Unicode на каждый символ отводятся два байта. Определите информационный объем следующего сообщения:

- 1) 1 бит; 2) 46 байт; 3) 96 байт; 4) 48 байт.

A2

В школе 32 компьютера размещены в кабинетах *A* и *B*. Сообщение "сломался компьютер из кабинета *A*" несет 3 бита информации. В кабинете *B* находится компьютеров:

- 1) 8; 2) 4; 3) 28; 4) 32.

A3

Для передачи секретного сообщения из 25 символов использовался код, состоящий из 12 букв. Все буквы кодируются одним и тем же (минимально возможным) количеством бит. Информационный объем такого сообщения равен:

- 1) 300 бит; 2) 75 бит; 3) 100 бит; 4) 25 бит.

A4

Для хранения целого числа со знаком в компьютере используется один байт. Сколько единиц содержит внутреннее представление числа -120 ?

- 1) 5; 2) 4; 3) 3; 4) 2.

A5

В четверичной системе счисления сумма чисел F_{16} и 14_8 равна:

- 1) 321; 2) 33; 3) 123; 4) 1101.

A6

В результате выполнения представленного алгоритма (рис. 18.1) переменные *A* и *C* примут значения:

- 1) $A = 1; C = -1$; 3) $A = 0; C = 0$;
2) $A = 1; C = 0$; 4) $A = 0; C = -1$.

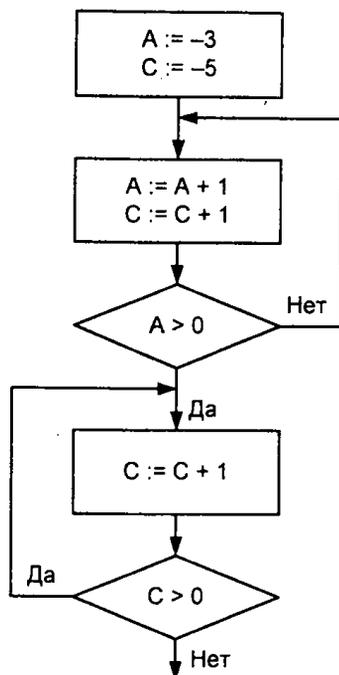


Рис. 18.1

A7

В результате выполнения фрагмента алгоритма:

Бейсик

```

DO WHILE n <> 0
PRINT (2 * (n MOD 10));
n = n \ 10
LOOP
    
```

(\ и Mod – операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно)

Паскаль

```

while n <> 0 do
begin
write (2 * (n mod 10))
n := n div 10;
end;
    
```

{div и mod – операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}

Алгоритмический язык

```

нц пока n <> 0
вывод (2 * mod (n, 10));
n := div (n, 10);
    
```

нц

| div и mod – функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

было напечатано число 104128. Значение переменной n перед выполнением фрагмента было равно:

- 1) 821401; 2) 12234; 3) 4132; 4) 4625.

A8

Фрагмент алгоритма:

нц для k от 1 до 3если $(x[2*k - 1] > x[2*k])$ то $S := x[2*k - 1]; x[2*k - 1] := x[2*k]; x[2*k] := S;$ всекц

полностью упорядочивает по возрастанию массив:

- 1) (74, 17, 12, 5, 15, 3); 3) (12, 15, 17, 3, 5, 74);
 2) (5, 15, 3, 12, 17, 74); 4) (5, 3, 15, 12, 17, 74).

A9

Высказывание:

$$(\sin(x) > 1) \vee (y = x \bmod 5) \wedge (x \operatorname{div} y < 3)$$

истинно при значениях:

- 1) $x = 3; y = 1;$ 3) $x = 5; y = 3;$
 2) $x = 8; y = 3;$ 4) $x = 1; y = 2.$

A10

Логическому выражению

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

равносильно выражение:

- 1) $\neg A;$ 3) $\neg A \vee B \vee \neg C;$
 2) $A \wedge \neg B \wedge C;$ 4) $\neg A \wedge B \wedge \neg C.$

A11

Дана таблица истинности выражения:

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

Какое выражение соответствует F ?

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

A12

Четырехзначных чисел, в записи которых встречается не более двух различных цифр, всего существует:

- 1) 567; 2) 546; 3) 556; 4) 576.

A13

Для кодирования букв "X", "Y", "Z", "W" решили использовать одно- и двухразрядные последовательные двоичные числа (от 0 до 11 соответственно). Если таким способом закодировать последовательность символов WYXZ и записать результат шестнадцатеричным кодом, то получится:

- 1) 148; 2) E8; 3) 310; 4) 3A.

A14

В непозиционной системе счисления, которая называется системой остатков (СО), в качестве оснований выбираются взаимно простые числа, например, $p_1 = 3$, $p_2 = 5$, $p_3 = 7$. При этом диапазон однозначного представления чисел равен произведению оснований $D = p_1 \cdot p_2 \cdot p_3 = 105$ (от 0 до 104). Любое число в этом диапазоне записывается остатками от целочисленного деления этого числа на выбранные основания. Например, число $A = 19$ запишется в системе с основаниями 3, 5, 7 так: $A_{(CO)} = (1, 4, 5)$. Укажите, какая из записей соответствует числу 3, записанному в указанной системе остатков:

- 1) (3, 0, 0); 2) (0, 3, 3); 3) (0, 2, 4); 4) (3, 3, 3).

A15

Каталог содержит файлы:

- а) z1.pas;
б) z21.pas;
в) z4.p;
г) z33.p;
д) zad.pas;
е) zom.pp.

При выделении файлов с использованием маски **z*.p??** список всех выделенных файлов будет таков:

- 1) а, б, д, е; 2) а, б, д; 3) б, г, д, е; 4) б, д.

A16

Представлена база данных "Расписание уроков":

№	День	N_урока	Кабинет	Предмет	Преподаватель	Класс
1	Пн	2	32	матем	Голубева	9а
2	Пн	2	21	физика	Иванова	8а
3	Вт	4	25	литер	Зайцев	8б
4	Вт	3	25	литер	Зайцев	8а
5	Чт	4	31	физика	Зайцева	10б
6	Пт	3	32	матем	Голубев	8а
7	Чт	2	41	химия	Панина	9а
8	Пн	1	28	матем	Петров	10а
9	Вт	1	41	химия	Панина	10б

Условию отбора $N_{\text{урока}} > 2$ И Класс $> '8a'$ удовлетворяют только записи №:

- 1) 1, 2, 3, 5, 7; 3) 3;
2) 3, 4, 5, 6; 4) 1, 3, 4, 6, 7.

A17

Цветной сканер имеет разрешение 256×512 точек/дюйм. Объем памяти, занимаемый отсканированным изображением размером 4×4 дюйма, составляет 6 Мбайт. Глубина представления цвета сканера в битах равна:

- 1) 96; 2) 3; 3) 24; 4) 6.

A18

Представлен фрагмент электронной таблицы, содержащей числа и формулы:

	B	C	D
69	5	10	
70	6	9	=СЧЁТ (B69:C70)
71			=СРЗНАЧ (B69:D70)

После перемещения содержимого ячейки C70 в ячейку C71 значение в ячейке D71 изменится по абсолютной величине на:

- 1) 2,2; 2) 2,0; 3) 1,05; 4) 0,8.

A19

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D	E
1	1	5			
2	0	1	1	1	1
3	=A2+\$A\$1	=A3*B2	=-C2+2*\$B\$1	=D2+A3*2	=E2+\$B\$1

После копирования диапазона ячеек A3:E3 в диапазон A4:E6 была построена диаграмма (график, рис. 18.2) по значениям столбцов диапазона ячеек B2:E6.

Значениям C2:C6 соответствует график:

- 1) A; 2) B; 3) C; 4) D.

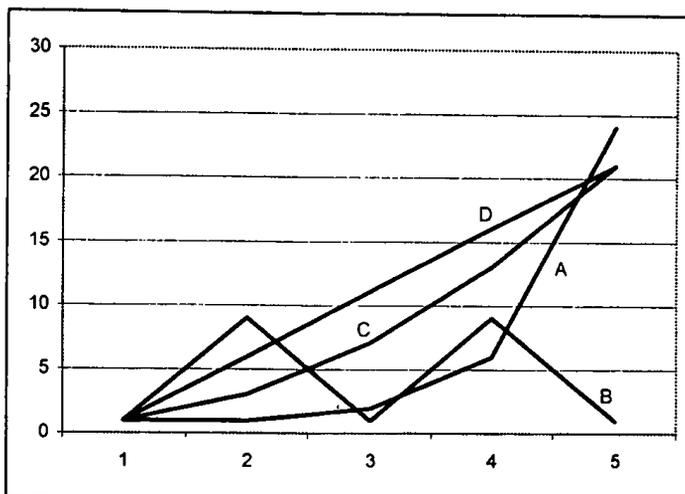


Рис. 18.2

A20

Исполнитель "ПРОЦЕССОР" имеет два регистра с именами A и B, в которых хранятся целые числа. В систему команд "ПРОЦЕССОРА" входят 6 команд:

?An	Ввод числа <i>n</i> в ячейку A
?Bn	Ввод числа <i>n</i> в ячейку B
!A	Вывод данных из ячейки A на экран
!B	Вывод данных из ячейки B на экран
+BA	К содержимому ячейки B добавить содержимое ячейки A и полученный результат поместить в ячейку B. Содержимое ячейки A остается неизменным
+AB	К содержимому ячейки A добавить содержимое ячейки B и полученный результат поместить в ячейку A. Содержимое ячейки B остается неизменным
-BA	Из содержимого ячейки B вычесть содержимое ячейки A и полученный результат поместить в ячейку B. Содержимое ячейки A остается неизменным
-AB	Из содержимого ячейки A вычесть содержимое ячейки B и полученный результат поместить в ячейку A. Содержимое ячейки B остается неизменным

После выполнения команд:

?A4 ?B6 -BA +AB -BA !A !B

на экран будут выведены числа:

- 1) 6 4; 2) 4 2; 3) 6 -4; 4) 4 -6.

ЧАСТЬ В. Ответ, как набор символов

B1

Укажите, сколько раз используется цифра 2 при записи чисел 13, 14, 15, ..., 22, 23 в системе счисления с основанием 3.

B2

Составьте таблицу истинности для функции:

$$F = \neg((A \leftrightarrow B) \rightarrow C) \vee \neg(A \wedge C),$$

в которой столбец значений аргумента A представляет собой двоичную запись числа 15, столбец значений аргумента B — числа 51, а столбец значений аргумента C — числа 85. Число в столбце записывается сверху вниз от старшего разряда к младшему. Переведите полученную двоичную запись значения функции F в десятичную систему счисления.

B3

Специализированный процессор работает с положительными целыми однобайтовыми числами. Он может выполнять две команды:

1. сдвиг числа вправо на один двоичный разряд.
2. прибавь 1.

Для заданного числа 38 задана последовательность команд 11222. Запишите полученный результат в десятичной системе счисления.

B4

Четырехкнопочный кодовый замок открывается, если выполняются следующие четыре условия:

1. Если не нажата кнопка 3, то нужно нажать кнопку 1 и не нажимать кнопку 4.
2. Если нажать кнопку 4, то нужно нажать кнопку 3 и не нажимать кнопку 2.
3. Неверно, что нужно нажать кнопку 2 или не нажимать кнопку 3, и все это притом, что не нажата кнопка 4.
4. Не нажимая кнопку 4, нужно нажать кнопку 1 или кнопку 3.

Запишите верный код замка. (В ответе перечислите номера кнопок, которые должны быть нажаты, в порядке возрастания, без пробелов.)

B5

Передача файла размером 640 Кбайт по каналу связи заняла 5 секунд. Скорость передачи данных по каналу связи равна _____ Кбит/с.

B6

Цепочки символов (строки) создаются по следующему правилу:

1. Нулевая строка состоит из одного символа — цифры 0.
2. Каждая из последующих цепочек создается такими действиями: в начало дважды подряд записывается предыдущая строка, далее инвертированная предыдущая строка (вместо цифры 0 цифра 1 и наоборот).

Вот первые строки, созданные по этому правилу:

- (0) 0
 (1) 001
 (2) 001001110
 (3) 001001110001001110110110001

Сколько раз встречается цифра 1 в строке с номером 6?

B7

Маской подсети называется 32-разрядное двоичное число, которое определяет, какая часть IP-адреса компьютера относится к адресу сети, а какая часть определяет IP-адрес компьютера в подсети. В маске подсети старшие биты, отведенные в IP-адресе компьютера для адреса сети, имеют значение 1; младшие биты, отведенные в IP-адресе компьютера для адреса компьютера в подсети, имеют значение 0. Например, маска подсети может иметь вид:

11111111 11111111 11100000 00000000 (255.255.224.0)

Это значит, что 19 старших битов в IP-адресе содержат адрес сети, оставшиеся 13 младших битов содержат адрес компьютера в сети.

Укажите, какие значения из представленных в таблице *не могут быть* маской подсети:

1	255.255.252.0
2	255.255.230.0
3	255.255.255.128
4	255.255.255.240

Запишите последовательно их номера, например, 134.

B8

В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

Если в запросе текст указан без кавычек, производится поиск текстов, в которых встречаются все слова запроса. Если текст записан в кавычках, производится поиск текстов, содержащих строго указанное словосочетание.

1	логическая операция
2	логическая операция импликация
3	"логическая операция импликация"
4	"логическая операция"

ЧАСТЬ С. Самостоятельные задания

C1

Требовалось написать программу, которая считает во введенной с клавиатуры строке, содержащей только буквы и пробелы (' '), количество слов, начинающихся и заканчивающихся одной и той же буквой. Слово из одной буквы также считается. Программист торопился и написал программу неправильно.

Паскаль

```

var
s: string;
a: char;
i, k: integer;
begin
  readln(s); k := 0;
  for i := 1 to length(s) do
    begin
      if (s[i]=' ') and (s[i+1]<>' ') then a := s[i+1];
      if (s[i]<>' ') and (s[i+1]=' ') then if s[i]=a then
        k := k + 1;
      if (s[i-1]=' ') and (s[i]<>' ') and (s[i+1]=' ') then
        k := k + 1;
    end;
  if k = 0 then
    write('таких слов нет')
  else
    write('k =', k);
end.

```

Бейсик

```

INPUT st$
K = 0
FOR i = 1 to LEN (st$)
  IF (MID$(st$,i,1)=" ") AND (MID$(st$,i,1) <> " ") THEN
    a$ = MID$(st$,i,1)
  IF MID$(st$,i,1)<>" " AND (MID$(st$,i,1) = " ") THEN
    IF MID$(st$,i,1) = a$ THEN k = k + 1
    IF MID$(st$,i-1,1) = " " AND MID$(st$,i,1) <> " " AND
    MID$(st$,i+1,1) = " " THEN k = k + 1
  ENDF
NEXT i
IF k = 0 THEN
  PRINT "таких слов нет"
ELSE
  PRINT k
ENDIF
END

```

Последовательно выполните 3 задания.

1. Приведите пример таких строк, при которых программа неверно решает поставленную задачу.
2. Укажите, какая часть программы является лишней.
3. Укажите, как нужно доработать программу, чтобы не было случаев неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

C2

Задан упорядоченный по возрастанию массив целых чисел. Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий вставить в этот массив введенное с клавиатуры число без нарушения упорядоченности.

C3

Два игрока играют в следующую игру. На клетчатой доске стоит фишка в точке с координатами $(2, 2)$. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: или в точку с координатами $(x + 1, y + 2)$, или в точку с координатами $(x, y + 3)$, или в точку с координатами $(x + 2, y + 1)$. Выигрывает игрок, после хода которого фигура окажется в точке с координатами $(8, 8)$. Если фигура окажется в точке $(8, y)$ или $(x, 8)$, считается, что победителя нет. Кто выигрывает при безошибочной игре: игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

C4

На вход программы подаются сведения о сдаче экзаменов абитуриентами, поступающими на одну специальность. В первой строке сообщается количество абитуриентов n , которое не превосходит 100, каждая из следующих n строк имеет следующий формат:

<Фамилия> <Инициалы> <баллы> <льгота>

где <Фамилия> — строка, состоящая не более чем из 20 символов, <Инициалы> — строка, состоящая из 4-х символов, <баллы> — через пробел три целых числа, соответствующие оценкам по стобалльной системе, <льгота> — число 0 или 1 (нет льгот или льгота есть). <Фамилия> и <Инициалы>, <Инициалы> и <баллы>, а также <Баллы> и <Льгота> разделены одним пробелом.

Пример входной строки:

Иванов П.К. 45 57 38 0

По плану должно быть зачислено 15 абитуриентов. Зачисление абитуриентов производится так:

- ♦ в первую очередь зачисляются абитуриенты, имеющие льготы, если они набрали более 30 баллов по каждому предмету;
- ♦ далее зачисляются абитуриенты в порядке убывания суммы баллов по трем предметам.

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить на экран одно из сообщений о зачислении абитуриентов:

- 1) "список успешно сформирован";
- 2) "имеются вакантные места";
- 3) "к зачислению рекомендуется более 15 абитуриентов".

ГЛАВА 19

Ответы и решения задач

К главе 1 "Кодирование текстовой информации, основные используемые кодировки"

- ◇ Задание A1-2008: № 4.
- ◇ Задание A1-2009: № 2.
- ◇ Задание A11-2010: № 2.
- ◇ Задание A12-2010: № 3.
- ◇ Задание B1-2009: 3.
- ◇ Задание B1-2010: 64.
- ◇ Задание 1.1: № 4.
- ◇ Задание 1.2: № 1.
- ◇ Задание 1.3: № 2.
- ◇ Задание 1.4: № 3.
- ◇ Задание 1.5: № 2.
- ◇ Задание 1.6: № 2.
- ◇ Задание 1.7: № 4.
- ◇ Задание 1.8: № 2.
- ◇ Задание 1.9: № 1.

К главе 2 "Вычисление информационного объема"

- ◇ Задание A2-2009: № 3.
- ◇ Задание A2-2010: № 2.
- ◇ Задание B7-2009: 40.
- ◇ Задание B7-2010: 1296.
- ◇ Задание 2.1: № 1.
- ◇ Задание 2.2: № 4.
- ◇ Задание 2.3: № 2.
- ◇ Задание 2.4: № 3.
- ◇ Задание 2.5: № 3.
- ◇ Задание 2.6: № 3.
- ◇ Задание 2.7: № 4.
- ◇ Задание 2.8: 30 000.
- ◇ Задание 2.9: 512.
- ◇ Задание 2.10: 4800.
- ◇ Задание 2.11: в 4 раза.
- ◇ Задание 2.12: 2400.

К главе 3 "Системы счисления"

- ◇ Задание A1-2010: № 2.
- ◇ Задание A3-2010: № 1.
- ◇ Задание A4-2010: № 4.
- ◇ Задание A11-2010: № 2.
- ◇ Задание B3-2010: 7.
- ◇ Задание 3.1: № 2.
- ◇ Задание 3.2: № 3.
- ◇ Задание 3.3: № 2.
- ◇ Задание 3.4: № 1.
- ◇ Задание 3.5: № 2.
- ◇ Задание 3.6: № 4.
- ◇ Задание 3.7: № 3.
- ◇ Задание 3.8: № 4.
- ◇ Задание 3.9: № 3.
- ◇ Задание 3.10: № 1.
- ◇ Задание 3.11: № 2.
- ◇ Задание 3.12: № 2.
- ◇ Задание 3.13: № 2.
- ◇ Задание 3.14: № 4.
- ◇ Задание 3.15: № 3.
- ◇ Задание 3.16: № 2.
- ◇ Задание 3.17: 4000.
- ◇ Задание 3.18: 7.
- ◇ Задание 3.19: 624.
- ◇ Задание 3.20: 7.
- ◇ Задание 3.21: 9, 27.

К главе 4 "Электронные таблицы и диаграммы"

- ◇ Задание А7-2010: № 1.
- ◇ Задание А8-2010: № 2.
- ◇ Задание А9-2010: № 3.
- ◇ Задание В4-2010: 30.
- ◇ Задание 4.1: № 3.
- ◇ Задание 4.2: № 4.
- ◇ Задание 4.3: № 2.
- ◇ Задание 4.4: № 2.
- ◇ Задание 4.5: № 4.
- ◇ Задание 4.6: № 4.
- ◇ Задание 4.7: № 2.
- ◇ Задание 4.8: № 1.
- ◇ Задание 4.9: № 4.
- ◇ Задание 4.10: № 3.
- ◇ Задание 4.11: № 3.
- ◇ Задание 4.12: № 4.
- ◇ Задание 4.13: № 4.
- ◇ Задание 4.14: № 4.
- ◇ Задание 4.15: № 4.
- ◇ Задание 4.16: 1.

К главе 5 "Алгебра логики"

- ◇ Задание А16-2009: № 4.
- ◇ Задание А17-2009: № 1.
- ◇ Задание А16-2010: № 3.
- ◇ Задание А17-2010: № 4.
- ◇ Задание 5.1: № 3.
- ◇ Задание 5.2: а) № 3; б) № 2; в) № 3.
- ◇ Задание 5.3: № 2.
- ◇ Задание 5.4: № 1.
- ◇ Задание 5.5: № 3.

К главе 6 "Закономерности"

- ◇ Задание А12-2009: 1.
- ◇ Задание В6-2008: 4.
- ◇ Задание В8-2009: BAAGFED.
- ◇ Задание В8-2010: AABAAB.
- ◇ Задание 6.1: № 4.
- ◇ Задание 6.2: 136.

К главе 7 "Базы данных"

- ◇ Задание А10-2009: № 4.
- ◇ Задание А14-2009: № 1.
- ◇ Задание А10-2010: № 2.
- ◇ Задание А14-2010: № 4.
- ◇ Задание 7.1: № 2.
- ◇ Задание 7.2: № 4.
- ◇ Задание 7.3: № 3.

К главе 8 "Файловая структура организации информации на компьютере"

- ◇ Задание А13-2009: № 3.
- ◇ Задание А13-2010: № 2.
- ◇ Задание 8.1: № 2.
- ◇ Задание 8.2: № 4.
- ◇ Задание 8.3: № 3.

К главе 9 "RGB-кодирование"

- ◇ Задание А13-2010: № 2.
- ◇ Задание 9.1: № 4.

К главе 10 "Исполнители"

- ◇ Задание В5-2009: 22111.
- ◇ Задание А18-2010: № 1.
- ◇ Задание В5-2010: 2.
- ◇ Задание 10.1: 211122.
- ◇ Задание 10.2: NO6.
- ◇ Задание 10.3: А1-В3-С5-А4-С5-Е6-Г7(или С7)-Е8.
- ◇ Задание 10.4: ПУПУУУПУП.
- ◇ Задание 10.5: 11011001.

К главе 11 "Алгоритмизация"

- ◇ Задание А6-2009: № 3.
- ◇ Задание А5-2010: № 1.
- ◇ Задание А6-2010: № 4.
- ◇ Задание В2-2010: 255.
- ◇ Задание 11.1: № 3.
- ◇ Задание 11.2: № 4.
- ◇ Задание 11.3: № 4.
- ◇ Задание 11.4: № 2.
- ◇ Задание 11.5: № 4.
- ◇ Задание 11.6: № 1.
- ◇ Задание 11.7: № 2.
- ◇ Задание 11.8: № 3.
- ◇ Задание 11.9: № 2.
- ◇ Задание 11.10: № 1.
- ◇ Задание 11.11: № 1.
- ◇ Задание 11.12: № 2.
- ◇ Задание 11.13: № 2.
- ◇ Задание 11.14: № 3.
- ◇ Задание 11.15: 19.
- ◇ Задание 11.16: 22.
- ◇ Задание 11.17: 3.

К главе 12 "Логические задачи"

- ◇ Задание В6-2009: СКМ.
- ◇ Задание В6-2010: АМВЕ.
- ◇ Задание 12.1: СКА.
- ◇ Задание 12.2: И1В2С3К4Е5, В1С2К3И4Е5.
- ◇ Задание 12.3: черный "Запорожец".
- ◇ Задание 12.4: БкзЧсзКчбСчкЗбс.
- ◇ Задание 12.5: АСБВ.
- ◇ Задание 12.6. А вот эту задачу, решите, пожалуйста, сами. Обычно у меня 2—3 человека из класса (не знающих, конечно, задачи) справляются за полчаса. Остальные, естественно, списывают.

К главе 13 "Интернет"

- ◇ Задание В9-2009: ГБВА.
- ◇ Задание В9-2010: ВБГА.
- ◇ Задание В10-2009: 1234.
- ◇ Задание В10-2010: 4321.
- ◇ Задание 13.1: 4132.
- ◇ Задание 13.2: 3412.
- ◇ Задание 13.3: 1432.

К главе 14 "Нахождение ошибок в готовых программах и их доработка"

- ◇ Задание С1-2009.
 - 1) Программист поторопился и не ограничил область рассмотрения значением $x \leq \pi/2$. Поэтому возможны значения x и y такие, при которых программа будет уверять, что точка принадлежит указанной области, хотя реально она ей не принадлежит. Это могут быть, на-

пример, значения: $x = 2$; $y = 0,95$. Программа в качестве результата выдаст ответ "принадлежит", а на деле — "не принадлежит".

2) Возможная доработка могла бы быть такой:

Бейсик

```
CLS
INPUT x, y
if y >= 1 AND x >= 0 AND y >= SIN(x) AND x <= 3.14/2 then
PRINT "принадлежит"
ELSE
PRINT "не принадлежит"
END IF
END
```

Паскаль

```
var x, y: real;
begin
  readln(x, y);
  if (y<=1) and (x>=0) and (y>=sin(x)) and (x<=3.14/2) then
    write('принадлежит')
  else
    write('не принадлежит')
end.
```

```
void main(void)
{ float x, y;
  scanf("%f%f", &x, &y);
  if (y <= 1)
  if (x >= 0)
  if (y >= sin(x))
  if (x <= 3.14/2)
    printf("принадлежит");
  else
    printf("не принадлежит");
}
```

◆ Задание С1-2010.

1) Пример: $x = -1$, $y = -3$. (Любая пара (x, y) , для которой выполняется: $x^2 + y^2 < 4$ или $x < -2$, или $y < 0$ и $y \leq -x$.)

2) Возможная доработка (Паскаль):

```
if (x*x+y*y >= 4) and (x >= -2) and (y <= -x) and (y >=0) then
  write('принадлежит')
else
  write('не принадлежит')
```

◆ Задание 14.1.

1) Пример: $x = 1, y = 0$ (Любая пара (x, y) , для которой выполняется: $y < x$ или $y < 0$.)

2) Возможная доработка (Паскаль, разбиение области на две части прямой $x = 0$):

```
if (y<=2-x*x) and (y>=0) and (x>=0) or (x<=0) and (y>=x) and (y<=2-x*x) then
  write('принадлежит')
else
  write('не принадлежит')
```

◆ Задание 14.2.

1) Очевидно, что при неопределенных заранее значениях $r1$ и $r2$ в программе вместо их значений будут заданы нули, поэтому программа будет неправильно работать, например, при $x = 3,5, y = 3,5$ (будет выдано сообщение "принадлежит").

Кроме того, значение переменной z всегда положительно (за исключением вышеописанного случая, когда оно принимает значение 0), поэтому первое рассматриваемое условие исполняться никогда (за исключением рассмотренного выше случая) не будет, и на экран не будет выведено никакого сообщения.

2) Возможная доработка такова. Надо, во-первых, задать значения $r1$ и $r2$. А во-вторых, связать все условия логической функцией AND.

Паскаль

```
const r1 = 2.5, r2 = 1.5;
var x, y, z: real;
begin
  readln(x,y);
  z := (x-3.5) * (x-3.5) + (y-3.5) * (y-3.5);
  if (r2*r2 <= z) and (r1*r1 >= z) and (y >= x) then
    write('принадлежит')
  else
    write('не принадлежит')
end.
```

Бейсик

```
r1 = 2.5
r2 = 1.5
INPUT x, y
z = (x-3.5) * (x-3.5) + (y-3.5) * (y-3.5)
IF r2^2 <= z AND r1^2 >= z AND y >= x THEN
PRINT "принадлежит"
ELSE
PRINT "не принадлежит"
ENDIF
END
```

К главе 15 "Написание простых программ"

◆ Задание C2-2009.

Сначала приведу словесное описание алгоритма на русском языке.

Считаем, что массив $A(30)$ уже создан и заполнен. Создан также массив в той же длины. Тогда будем в цикле поэлементно перебирать массив a . Если элемент меньше нуля, то он переписывается в массив b с обратным знаком, если же он больше либо равен нулю, то переписывается в массив b без изменений.

В качестве примера правильного и эффективного алгоритма приведу фрагмент программы, разработанный мною.

Бейсик

```
'объявляем массивы A и B объемом 30 элементов
DIM a(30), b(30)
'очищаем экран — чтобы красиво было
CLS
'подключаем оператор случайных чисел
RANDOMIZE TIMER
'организуем цикл по заполнению массива случайными числами от -20 до 19 включительно
FOR i = 1 TO 30
a(i) = INT(RND(1) * 40) - 20
'выводим для контроля получившийся массив на экран
PRINT a(i);
NEXT i
'пропускаем пустую строку
PRINT
'организуем цикл, переписывания из массива A в массив B
FOR i = 1 TO 30
IF a(i) < 0 THEN
b(i) = -a(i)
ELSE
b(i) = a(i)
END IF
NEXT i
'выведем на экран получившийся массив
FOR I = 1 to 30
PRINT b(i);
NEXT i
END
```

Паскаль

```
program modul;
{подключаем модуль очистки экрана}
uses crt;
{блок описания переменных}
var
  a: array [1..20 ] of integer;
  b: array [1..20 ] of integer;
  i: integer;
```

```

begin
  {очищаем экран - чтобы красиво было}
  clrscr;
  {подключаем оператор случайных чисел}
  randomize;
  {организуем цикл по заполнению массива случайными числами от -20 до 19 включительно}
  for i := 1 TO 30 do
  begin
    a[i] := random(40) - 20;
    {выводим для контроля получившийся массив на экран}
    write (a[i], ' ');
  end;
  {пропускаем пустую строку}
  writeln; writeln;
  for i = 1 to 30 do
  begin
    if a[i] < 0 then
      b[i] := -a[i]
    else
      b[i] := a[i];
    write (b[i], ' ');
  end;
end.

```

◆ Задание C2-2010.

Паскаль

```

min := 100;
for i := 1 to N do
  if (a[i] >= 20) and (a[i] < min) then
    min := a[i];
writeln(min);

```

C++

```

min = 100;
for (i = 0; i < N; i++)
  if (a[i] >= 20 && a[i] < min)
    min = a[i];
printf("% d", min);

```

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

```

Естественный язык

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.

◆ Задание 15.1.

Паскаль	Бейсик
x:=0;	X = 0
m:=0;	M = 0
for i:=1 to N do	FOR I = 1 TO N
if (a[i] mod 10=7) then begin	IF A(I) MOD 10 = 7 THEN
x:=x+a[i];	X = X + A(I)
m:=m+1;	M = M + 1
end;	ENDIF
s:=x/m;	NEXT I
writeln(s);	S = X / M
	PRINT S

◆ Задание 15.2.

```

Бейсик
'объявляем массив А объемом 100 элементов
DIM a(100)
'очищаем экран – чтобы красиво было
CLS
'подключаем оператор случайных чисел
RANDOMIZE TIMER
по заполнению массива случайными числами от 0 до 500 включительно
' и нахождения суммы его элементов
S = 0
FOR i = 1 TO 100
a(i) = INT(RND(1) * 501)
'выводим для контроля получившийся массив на экран
PRINT a(i);
S = S + a(i)
NEXT i
'пропускаем пустую строку
PRINT
'вычисляем среднее арифметическое элементов массива
SA = S/100
N = 0
'организуем цикл подсчета кол-ва элементов, меньших среднего арифметического
FOR i = 1 TO 100
IF a(i) < SA THEN
N = N + 1
END IF
NEXT i
'выведем на экран получившийся результат
PRINT "Кол-во элементов, меньших среднего арифметического ="; N
END
    
```

Паскаль

```

program modul;
{подключаем модуль очистки экрана}
uses crt;
{блок описания переменных}
var
  a: array [1..100 ] of integer;
  sa: real;
  I, sum, n: integer;
begin
  {очищаем экран – чтобы красиво было}
  clrscr;
  {подключаем оператор случайных чисел}
  randomize;
  {организуем цикл по заполнению массива случайными числами от 0 до 500 включительно}
  sum := 0;
  for i := 1 TO 100 do
  begin
    a[i] := random(501);
    {выводим для контроля получившийся массив на экран}
    {и находим сумму элементов массива}
    sum := sum + a[i];
    write (a[i], ' ');
  end;
  {пропускаем пустую строку}
  writeln; writeln;
  {вычисляем среднее арифметическое}
  sa := sum/100;
  {организуем цикл по подсчету кол-ва элементов, меньших среднего арифметического}
  n := 0;
  for i = 1 to 100 do
  if a[i] < sa then
    n := n + 1;
  {выводим результат}
  writeln('Количество элементов, меньших среднего арифметического =', n);
end.

```

◆ Задание 15.3.

Сначала попробую изложить на естественном языке.

Буду перебирать элементы массива, начиная с первого, до тех пор, пока не найдется рост, больший либо равный 180 см. Запомню номер этого элемента массива в переменной *i*. В переменную *min* запишу значение этого роста.

Теперь в цикле по переменной *j* (изменяющей свое значение от *i* до *n* с шагом 1) буду сравнивать каждое последующее значение роста со значением переменной *min* в логической связке AND с условием, что рост должен быть больше либо равен 180 см. Если найдется такой рост, который больше либо равен 180 см и при этом меньше *min*, то именно его значение и записываю в переменную *min*. По окончании цикла вывожу на экран значение *min*.

Теперь приведу пример правильной и эффективной программы на языках Паскаль и Бейсик.

Паскаль

```

const
  N = 30
var
  A: array [1..N] of integer;
  i, j, min: integer;
begin
  for i := 1 to N do
    readln(a[i]);
  i := 1;
  while a[i] < 180 do
    i := i + 1;
  min := a[i];
  for j := i to N do
    if (a[j] < min) and (a[j] >= 180) then min := a[j];
  writeln ('Минимальный рост баскетболиста =', min);
  readln
end.

```

Бейсик

```

N = 30
DIM A(N) AS INTEGER
DIM I, J, MIN AS INTEGER
FOR I = 1 TO N
INPUT A(I)
NEXT I
I = 1
WHILE A(I) < 180
I = I + 1
WEND
MIN = A(I)
FOR J = I TO N
IF A(J) < MIN AND A(J) >= 180 THEN MIN = A(J)
NEXT k
PRINT "Минимальный рост баскетболиста=";min
END

```

◆ Задание 15.4.

Бейсик

```

N = 30
DIM A(N), B(N) AS INTEGER
DIM I, J, R AS INTEGER
CLS

```

```

' заполнение массива с клавиатуры
FOR I = 1 TO N
INPUT A(I)
PRINT A(I);
NEXT I
PRINT
PRINT
' цикл поиска нечетного элемента и следующего за ним четного
' с последующей их записью в новый массив наоборот
FOR I = 1 TO N - 1
IF A(I) MOD 2 <> 0 THEN
    IF A(I + 1) MOD 2 = 0 THEN
        B(I) = A(I + 1)
        B(I + 1) = A(I)
        I = I + 1
    ELSE
        B(I) = A(I)
    END IF
ELSE
    B(I) = A(I)
END IF
    IF I = N - 1 THEN
        B(N) = A(N)
    END IF
NEXT I
' вывод получившегося массива
FOR I = 1 TO N
    PRINT B(I);
NEXT
END

```

А опытный программист легко текст Бейсика в Паскаль переведет. Если будут трудности, пишите.

К главе 16 "Нахождение выигрышной стратегии"

◆ Задание С3-2008.

Выигрывает второй игрок.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы (табл. 19.1), где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучках соответственно.

Таблица 19.1 содержит все возможные варианты ходов первого игрока. Из таблицы можно видеть, что при любом ходе первого игрока у второго имеется такой ход, который приводит к победе.

◆ Задание С3-2009.

Выигрывает второй игрок. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны координаты фишки на каждом этапе игры (табл. 19.2).

Таблица 19.2 содержит *все возможные* варианты ходов первого игрока. Из нее видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.

Таблица 19.1

Начало	1-й ход	2-й ход	3-й ход	4-й ход	Пояснение
	1-й игрок (все варианты хода)	2-й игрок (выигрышный ход)	1-й игрок (все варианты хода)	2-й игрок (один из вариантов)	
1, 2	3, 2	3, 4	9, 4	18, 4	Второй игрок выигрывает на четвертом ходу после любого ответа первого игрока, например, утроив число камней в самой большой куче
			5, 4	15, 4	
			3, 12	3, 36	
			3, 6	3, 18	
	1, 4		3, 4		Те же варианты третьего-четвертого ходов
1, 6		1, 18		Второй игрок выигрывает ответным ходом	

Таблица 19.2

Стартовая позиция	1-й ход	2-й ход	3-й ход	4-й ход
	1-й игрок (все варианты хода)	2-й игрок (выигрышный ход)	1-й игрок (все варианты хода)	Выигрышный ход (один из вариантов)
5, 2	5, 6	8, 6	11, 6	14, 6
			8, 9	11, 9
			8, 10	11, 10
	5, 5	8, 5	11, 5	14, 5
			8, 8	11, 8
			8, 9	11, 9
	8, 2	8, 5 или 8, 6	Те же варианты 3-го и 4-го ходов	

◆ Задание СЗ-2010.

Выигрывает первый игрок, своим первым ходом он должен поставить фишку в точке с координатами (1, -1). Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке координаты фишки на каждом этапе игры (табл. 19.3).

Таблица 19.3 содержит *все возможные* варианты ходов второго игрока. Из нее видно, что при любом ответе второго игрока у первого имеется ход, приводящий к победе.

◆ Задание 16.1.

Выигрывает второй игрок. Для доказательства рассмотрим неполное дерево игры (табл. 19.4). Из табл. 19.4 видно, что при любом первом ходе первого игрока у второго игрока имеется выигрывающий ход.

◆ А для заданий 16.2—16.10 у меня для вас решений нет. Мне интересны ваши.

Таблица 19.3

1-й ход	2-й ход	3-й ход	4-й ход	5-й ход
Позиция после первого хода	2-й игрок (все варианты хода)	1-й игрок (выигрышный ход)	2-й игрок (все варианты хода)	1-й игрок (один из вариантов)
1, -1	3, 1	5, 3	8, 3	11, 3
			5, 7	8, 7
			7, 5	10, 5
	1, 3	4, 3	7, 3	10, 3
			4, 7	7, 7
			6, 5	9, 5
	4, -1	4, 3	Те же варианты 3-го и 4-го ходов	

Таблица 19.4

Стартовая позиция	1-й ход	2-й ход	3-й ход	4-й ход
	1-й игрок (все варианты хода)	2-й игрок (выигрывающий ход)	1-й игрок (все варианты хода)	2-й игрок (выигрышный ход)
-3, 2	-3, 6 (x, y + 4)	2, 6 (x + 5, y) (расстояние до центра ≈ 6,32)	7, 6 (x + 5, y) (расстояние до центра ≈ 9,21)	12, 6 (x + 5, y) (расстояние до центра ≈ 13,14)
			2, 10 (x, y + 4) (расстояние до центра ≈ 10,19)	7, 10 (x + 5, y) (расстояние до центра ≈ 12,21)
			5, 9 (x + 3, y + 3) (расстояние до центра ≈ 10,77)	8, 12 (x + 3, y + 3) (расстояние до центра ≈ 14,42)
	2, 2 (x + 5, y)	5, 5 (x + 3, y + 3) (расстояние до центра ≈ 7,07)	10, 5 (x + 5, y) (расстояние до центра ≈ 11,18)	15, 5 (x + 5, y) (расстояние до центра ≈ 15,81)
			5, 9 (x, y + 4) (расстояние до центра ≈ 10,29)	12, 9 (x + 5, y) (расстояние до центра ≈ 15,00)
			8, 8 (x + 3, y + 3) (расстояние до центра ≈ 11,31)	11, 11 (x + 3, y + 3) (расстояние до центра ≈ 15,56)
	0, 5 (x + 3, y + 3)	5, 5 (x + 5, y) (расстояние до центра ≈ 6,32)	10, 5 (x + 5, y) (расстояние до центра ≈ 11,18)	15, 5 (x + 5, y) (расстояние до центра ≈ 15,81)
			5, 9 (x, y + 4) (расстояние до центра ≈ 10,29)	12, 9 (x + 5, y) (расстояние до центра ≈ 15,00)
			8, 8 (x + 3, y + 3) (расстояние до центра ≈ 11,31)	11, 11 (x + 3, y + 3) (расстояние до центра ≈ 15,56)

К главе 17 "Написание сложных программ"

◆ Задание С4-2009.

Программа верно читает входные данные, не запоминая их все, а сразу подсчитывая в массиве, хранящем 99 целых чисел согласно номерам школ, количество участников олимпиады из каждой школы. Затем с использованием ненулевых элементов этого массива ищется минимальный элемент, затем распечатываются номера соответствующих школ. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая.

Далее приведены примеры такого алгоритма на Бейсике и на Паскале.

Бейсик

```
'Объявляем массив количества участников от школ
DIM nc(99);
' Зачищаем экран от всякого ненужного
CLS
' Вводим количество учащихся
INPUT N
WHILE RIGHT$(d$, 1) <> "."
INPUT k$
d$ = d$ + k$
WEND
' Зная коды строчных латинских букв (от 97 до 122),
' производим подсчет таковых в нашей строке d$
n = LEN(d$): r = 0
DIM f(122, 26), bu$(26), num(26)
FOR k = 1 TO 26
FOR j = 97 TO 122
f(j, k) = 0
FOR i = 1 TO n
test$ = MID$(d$, i, 1)
IF test$ = CHR$(j) THEN f(j, k) = f(j, k) + 1
NEXT i
k = k + 1
NEXT j
NEXT k
' Формируем массив bu$ встречающихся в тексте букв
' и массив num - количества этих букв в строке.
' Кроме того, выводим на экран промежуточные результаты
k = 1: r = 1
FOR j = 97 TO 122
IF f(j, k) <> 0 THEN PRINT CHR$(j), f(j, k): bu$(r) = CHR$(j): num(r) = f(j, k):
r = r + 1
k = k + 1
NEXT j
PRINT "r="; r
' Сортируем оба массива по убыванию частоты символов
FOR i = 1 TO r - 1
FOR j = 1 TO r - 2
```

```

IF num(j) < num(j + 1) THEN SWAP num(j), num(j + 1): SWAP bu$(j), bu$(j + 1)
NEXT j
NEXT i
' Формируем выходную строку и выводим ее на экран
PRINT
exit$ = ""
FOR i = 1 TO r - 1
exit$ = exit$ + bu$(i)
'PRINT bu$(i), num(i)
NEXT i
PRINT exit$
END

```

Паскаль

```

program uchastniki;
var
  nc: array[1..99] of integer;
  p: 1..99;
  c: char;
  i, k, N, min: integer;
begin
  readln(N);
  for i := 0 to 99 do nc[i] := 0;
  for i := 1 to N do
    begin
      repeat
        read(c)
      until c = ' '; {считана фамилия}
      repeat
        read(c)
      until c = ' '; {считаны инициалы}
      readln(p);
      nc[p] := nc[p] + 1;
    end;
  min := N;
  for i := 1 to 99 do
    if nc[i] > 0 then
      begin
        if nc[i] < min then min := nc[i];
      end;
  for i := 1 to 99 do
    if nc[i] = min then
      writeln(i);
  readln
end.

```

◆ Задание С4-2010.

```
var
min, ans: array[92..98] of integer;
c: char;
i, k, N, b: integer;
begin
  for i := 92 to 98 do
    begin
      min[i] := 3001; {допустимо и другое число, >3000}
      ans[i] := 0;
    end;
  readln(N);
  for i := 1 to N do
    begin
      repeat
        read(c);
      until c = ' '; {считана компания}
      repeat
        read(c);
      until c = ' '; {считана улица}
      readln(k, b);
      if min[k] > b then
        begin
          min[k] := b;
          ans[k] := 1
        end else
          if min[k] = b then ans[k] := ans[k] + 1;
      end;
      {если бензина какой-то марки не было, ans[i] осталось рав}
      writeln(ans[92], ' ', ans[95], ' ', ans[98])
    end.
end.
```

◆ Задание 17.1.

Решение от авторов задачи.

Пример правильной и эффективной программы на Паскале:

```
Const d: array[1..12] of integer = (31, 28, 31, 30, 31, 30,
var tm: array[1..12] of real;
    m: array[1..365] of 1..12;
    data: string[5];
    max, ty, t: real;
    i: integer;
begin
  assign(input, 'meteo.dat');
  reset(input);
  for i := 1 to 12 do
    tm[i] := 0;
  ty := 0;
```

```

for i := 1 to 365 do
begin
  readln(data, t);
  m[i] := (ord(data[4]) - ord('0')) * 10
          +ord(data[5]) - ord('0');
  tm[m[i]] := tm[m[i]] + t;
  ty := ty + t;
end;
for i := 1 to 12 do
  tm[i] := tm[i]/d[i];
ty := ty/365;
max := 0;
for i := 1 to 12 do
  if abs(tm[i] - ty) > max then
    max := abs(tm[i] - ty);
writeln(ty:0:2);
for i := 1 to 12 do
  if abs(tm[i] - ty) = max then
    writeln(m[i], ' ', tm[m[i]]:0:2, ' ', tm[m[i]] - ty:0:2)
end.

```

◆ Задание 17.2.

Если вы добрались до задач группы С4 — вы опытный ЕГЭист, надеюсь вам интересно самому решить эту задачу.

К главе 18 "Вариант ЕГЭ повышенной сложности"

- ◆ Задание А1: № 3.
- ◆ Задание А2: № 3.
- ◆ Задание А3: № 3.
- ◆ Задание А4: № 4.
- ◆ Задание А5: № 3.
- ◆ Задание А6: № 2.
- ◆ Задание А7: № 4.
- ◆ Задание А8: № 4.
- ◆ Задание А9: № 2.
- ◆ Задание А10: № 1.
- ◆ Задание А11: № 4.
- ◆ Задание А12: № 4.
- ◆ Задание А13: № 4.
- ◆ Задание А14: № 2.
- ◆ Задание А15: № 2.
- ◆ Задание А16: № 3.
- ◆ Задание А17: № 3.
- ◆ Задание А18: № 4.
- ◆ Задание А19: № 2.
- ◆ Задание А20: № 3.

- ◆ Задание В1: 13.
- ◆ Задание В2: 250.
- ◆ Задание В3: 12.
- ◆ Задание В4: 13.
- ◆ Задание В5: 1024 Кбит/с.
- ◆ Задание В6: 364.
- ◆ Задание В7: 2.
- ◆ Задание В8: 3241.
- ◆ Задание С1.

Начнем с того, что сами приведенные варианты программ на Паскале и Бейсике не совпадают (это касается первого условного оператора). Поэтому может получиться по-разному. Но спишем все на случайность, опечатку.

Элементы ответа: программа не обнаруживает искомые слова в строках, содержащих таковые в начале или конце, например:

КАК ТАК — ГДЕ ОНО

Кроме того, слова, состоящие из одного слова, считаются дважды. Для следующей строки ответом будет 2:

ВОЙНА И МИР

1) Лишняя часть:

```
if (s[i-1]=' ') and (s[i]<>' ') and (s[i+1]=' ') then
  k := k+1;
```

2) Возможная доработка.

Один из вариантов может быть такой: после запроса строки дополнить ее пробелами в начале и конце и удалить лишнюю часть.

Пример программа на языке Паскаль

```
var
s: string;
a: char;
i, k: integer;
begin
  readln(s); k := 0;
  s := ' ' + s + ' '
  for i:=1 to length(s) do
  begin
    if (s[i]=' ') and (s[i+1]<>' ') then a:= s[i+1];
    if (s[i]<>' ') and (s[i+1]=' ') then if s[i]=a then
      k := k + 1;
  end;
  if k = 0 then
    write('таких слов нет')
  else
    write('k =', k);
end.
```

Пример программы на языке Бейсик

```

CLS
DIM st, a, b AS STRING
DIM k, i AS INTEGER
INPUT st$
k = 0
a$ = MID$(st$, 1, 1)
FOR i = 1 TO LEN(st$)
IF (MID$(st$, i, 1) <> " " AND MID$(st$, i + 1, 1) = " ") OR (MID$(st$, i, 1) <> " " AND
i = LEN(st$)) THEN b$ = MID$(st$, i, 1): IF a$ = b$ THEN k = k + 1
IF MID$(st$, i, 1) = " " AND MID$(st$, i + 1, 1) <> " " THEN a$ = MID$(st$, i + 1, 1)
NEXT i
IF k = 0 THEN
PRINT "таких слов нет"
ELSE
PRINT k
END IF
END

```

◆ Задание C2.

Самый простой способ — дописать введенное число в конец исходного массива и отсортировать его заново методом пузырька, например, или методом обмена. С этим, надеюсь, вы справитесь сами.

А вот с этим способом попробуйте разобраться: находим номер элемента массива, вслед за которым должен встать введенный, и затем сдвигаем все остальные элементы на единицу вправо.

Пример программы на языке Паскаль

```

const n = 30;
var a: array [1..n+1] of integer;
s, i, j: integer;
begin
  readln (s);
  i := 1;
  while (s > a[i]) and (i <= n) do
    i := i + 1;
  for j := n downto i do a[j + 1] := a[j];
  a[i] := s;
end.

```

Пример программы на языке Бейсик

```

N = 30
DIM s, i, j, a(N+1) AS INTEGER
INPUT s
i = 1
WHILE (s > a(i)) and (i <= N)
i = i + 1
WEND

```

```
FOR j = n TO j STEP -1
a(j+1) = a(j)
NEXT j
a(j) = s
END
```

◆ Задание С3.

Ну вот, собственно, у меня сложилось впечатление, что по правилам этой игры — при игре не в поддавки — никто не выиграет. Это как в детские "крестики-нолики" до трех, там при правильной игре всегда ничья. Так и тут.

Для доказательства рассмотрим дерево игры, оформленное в виде таблицы (табл. 19.5), где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют координатам фигуры на каждом ходе игры.

Таблица 19.5

Стартовая позиция	1-й ход	2-й ход	3-й ход	4-й ход
	1-й игрок (1-й вариант хода)	2-й игрок (все варианты хода)	1-й игрок (все варианты хода)	2-й игрок (один из вариантов)
2, 2	3, 4 (x + 1, y + 2)	4, 6 (x + 1, y + 2)	5, 8	выигравших нет
			4, 9	выигравших нет
			6, 7 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки	
		3, 7 (x, y + 3)	4, 9	выигравших нет
			3, 10	выигравших нет
			5, 8	выигравших нет
		5, 5 (x + 2, y + 1)	6, 7 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки	
			5, 8	выигравших нет
			7, 6 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки	

Второй вариант первого хода первого игрока приведен в табл. 19.6.

Третий вариант первого хода первого игрока — табл. 19.7.

Таблица 19.7 содержит все возможные варианты ходов первого и второго игрока. Из нее видно, что при любых ходах игроков, при их стремлении не дать выиграть другому, игра всегда закончится ситуацией "победителей нет".

Таблица 19.6

Стартовая позиция	1-й ход	2-й ход	3-й ход	4-й ход	
	1-й игрок (2-й вариант хода)	2-й игрок (все варианты хода)	1-й игрок (все варианты хода)	2-й игрок (один из вариантов)	
2, 2	2, 5 ($x, y + 3$)	3, 7 ($x + 1, y + 2$)	4, 9	выигравших нет	
			3, 10	выигравших нет	
			5, 8	выигравших нет	
		2, 8 ($x, y + 3$)	выигравших нет		
		4, 6 ($x + 2, y + 1$)	5, 8	выигравших нет	
			4, 9	выигравших нет	
			6, 7 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки		

Таблица 19.7

Стартовая позиция	1-й ход	2-й ход	3-й ход	4-й ход	
	1-й игрок (3-й вариант хода)	2-й игрок (все варианты хода)	1-й игрок (все варианты хода)	2-й игрок (один из вариантов)	
2, 2	4, 3 ($x + 2, y + 1$)	5, 5 ($x + 1, y + 2$)	6, 7 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки		
			5, 8	выигравших нет	
			7, 6 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки		
		4, 6 ($x, y + 3$)	6, 7 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки		
			4, 9	выигравших нет	
			5, 8	выигравших нет	
		6, 4 ($x + 2, y + 1$)	7, 6 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки		
			6, 7 так ходить нельзя — это ход для выигрыша второго — а мы играем не в поддавки		
			8, 7	выигравших нет	

◆ Задание С4.

Предлагаю вот такой вариант:

Пример программы на языке Паскаль

```

program C4;
  Var
    a:array [1..100] of Integer;
    s1: String;
    s2: string[2];
    sum, n, i, j, lg ,nelg, l, r, bl, p, rr, max: integer;
    ch: char;
begin
  readln(N);
  lg := 0; nelg := 0;
  for i := 1 to n do
  begin
    readln(s1);
    p := pos(' ', s1);
    s1 := copy(s1, p+1, length(s1)-p);
    p := pos(' ', s1);
    s1 := copy(s1, p+1, length(s1)-p);
    ch := s1[10];
    l := 0; sum := 0;
    for j := 1 to 3 do
    begin
      s2 := copy(s1, 1, 2);
      s1 := copy(s1, 4, length(s1) - 3);
      val(s2, bl, r);
      sum := sum + bl;
      if bl > 30 then l := l + 1;
    end;
    if l = 3 then
    begin
      if ch = '0' then
      begin
        nelg := nelg + 1;
        a[nelg] := sum;
      end
      else
        lg := lg + 1;
    end;
  end;
  if lg > 15 then
    writeln('рекомендовано >15')
  else
    if lg + nelg < 15 then
      writeln('есть места')

```

```
)  
_____  
else  
begin  
  if (lg = 15) or (lg + nelg = 15) then  
    writeln('список успешно сформирован')  
  else  
    begin  
      sum := 15 - lg;  
      for i := 1 to sum+1 do  
        begin  
          max := i;  
          for j := 2 to nelg do if a[j] > a[max] then max := j;  
          rr := a[i]; a[i] := a[max]; a[max] := rr;  
        end;  
      if a[Sum] = a[sum + 1] then writeln('рекомендовано >15')  
      else writeln('список успешно сформирован')  
    end;  
end;  
end.
```

Рекомендуемые интернет-ресурсы по ЕГЭ

- ◇ Министерство образования и науки Российской Федерации
<http://www.mon.gov.ru/>
<http://2379763.495.su/>
- ◇ Федеральный институт педагогических измерений
<http://www.fipi.ru/>
- ◇ Сайт информационной поддержки ЕГЭ
<http://www.ege.ru/>
- ◇ Российский общеобразовательный портал
<http://www.school.edu.ru/>
- ◇ Московское образование
<http://www.mosedu.ru/>
- ◇ Online-тестирование
http://www.klyaksa.net/test_online/
- ◇ Электронный репетитор
http://www.e-mentor.ru/download/ege_inform.shtml
- ◇ Московский институт открытого образования
<http://schools.keldysh.ru/faculty-it/ege/ege.htm>
- ◇ Задачи для подготовки к ЕГЭ
<http://www.aktanish.ru/edu/school/aktsh3/info/ege.htm>
- ◇ Молодежный форум по образованию
<http://forum.allsoch.ru/>
- ◇ Online-тест на "ЕГЭизм". Проверься!
<http://www.ht.ru/on-line/tests/egeism.php>
- ◇ Сайт полезных ссылок по ЕГЭ
http://www.eltarea.ru/rb-topic_t_194.htm
- ◇ Каталог интернет-ресурсов для абитуриента
<http://www.abiturcenter.ru/>
- ◇ Федеральный портал "Российское образование"
<http://www.edu.ru/>
- ◇ Информатика и ИКТ в образовании
<http://www.rusedu.info/Article92.html>
- ◇ Учебно-методический комплекс по информатике и ИКТ
<http://www.ito.su/2004/html/4614.html>
- ◇ Рекомендации XII Международной конференции-выставки "Информационные технологии в образовании" ("ИТО-2002")
<http://ito.edu.ru> , <http://ito.bitpro.ru>

- ◇ Обсуждаем проект стандарта по информатике и задания ЕГЭ
<http://2003.pedsovet.alledu.ru/forum/636/416/5320/1>
- ◇ Министерство образования и науки РФ
<http://www.ed.gov.ru/>
- ◇ Электронные пособия по информатике
<http://www.lbz.ru/index.php?div=downloads>
- ◇ Ресурсный центр. Методика. На сайте:
<http://onmcsso.narod.ru/metod.htm>
- ◇ Информационные технологии в образовании
<http://www.bolgar.info/index.php?module=News&catid=&topic=5&allstories=1&MDPROSID=af199c1c98530e4d7c94a6767c0cf68e>
- ◇ Дистанционное образование
http://sputnik.mto.ru/Docs_26/Ped.jurnal/Vio/Vio_15/cd_site/Articles/art_1_17.htm
- ◇ Все образование: каталог ссылок
<http://catalog.alledu.ru/>
- ◇ Все образование: информатика
<http://www.catalog.alledu.ru/predmet/info/>
- ◇ Федеральное государственное учреждение "Государственный научно-исследовательский институт информационных технологий и телекоммуникаций" (до 1999 г. центр "Информика")
<http://www.informika.ru/>
- ◇ "Учительская газета"
<http://www.ug.ru>
- ◇ "Первое сентября"
<http://www.1september.ru/ru/first.htm>
- ◇ "Курьер образования"
<http://www.courier.com.ru/>
- ◇ Дистанционные обучающие программы
<http://www.eduland.ru>
- ◇ Наука и образование
<http://edu.rin.ru/index.html>
- ◇ ЕГЭ по информатике
<http://www.156.ru/teachers-room/information/exam/exam.htm>
- ◇ Задачи для проведения ЕГЭ по информатике
<http://som.fio.ru/item.asp?id=10009004>
- ◇ Подготовка к ЕГЭ по информатике
<http://kpolyakov.narod.ru/school/ege.htm>

Литература

1. Босова Л. Л., Босова А. Ю., Коломенская Ю. Г. Занимательные задачи по информатике. — М.: Бином, 2005.
2. Гусева И. Ю. ЕГЭ. Информатика. Раздаточный материал тренировочных тестов. — СПб.: Тригон, 2009.
3. Ефимова О., Морозов В., Шафрин Ю. Курс компьютерной технологии. — М.: АБФ, 1998.
4. Задачник-практикум / Под ред. И. Г. Семакина, Е. К. Хеннера. — М.: Лаборатория базовых знаний, 2008.
5. Златопольский Д. М. Интеллектуальные игры по информатике. — СПб.: БХВ-Петербург, 2008.
6. Молодцов В. А., Рыжикова Н. Б. Информатика: тесты, задания, лучшие методики. — Ростов-на-Дону: Феникс, 2008.
7. Сафронов И. К. Задачник-практикум по информатике. — СПб.: БХВ-Петербург, 2002.
8. Соколова О. Л. Поурочные разработки по информатике. — М.: ВАКО, 2006.
9. Угринович Н. Д., Босова Л. Л., Михайлова Н. И. Практикум по информатике и информационным технологиям. — М.: Лаборатория базовых знаний, 2008.
10. Щикот С. Е., Крамаров С. О., Перепелкин В. В. Комплексные тестовые упражнения по информатике. — Ростов-на-Дону: Феникс, 2005.
11. Якушкин П. А., Крылов С. С. ЕГЭ 2010. Информатика. Сборник экзаменационных заданий. — М.: ЭКСМО, 2010.
12. Ярцева О. В., Цикина Е. Н. Информатика. ЕГЭ-2009. Самые новые задания. — М.: АСТ, 2009.

Оглавление

Предисловие	3
ГЛАВА 1. Кодирование текстовой информации, основные используемые кодировки.....	5
ГЛАВА 2. Вычисление информационного объема.....	13
ГЛАВА 3. Системы счисления	21
ГЛАВА 4. Алгебра логики.....	34
ГЛАВА 5. Электронные таблицы и диаграммы.....	48
ГЛАВА 6. Закономерности.....	57
ГЛАВА 7. Базы данных.....	62
ГЛАВА 8. Файловая структура организации информации на компьютере	69
ГЛАВА 9. RGB-кодирование	73
ГЛАВА 10. Исполнители	75
ГЛАВА 11. Алгоритмизация	82
ГЛАВА 12. Логические задачи	100
ГЛАВА 13. Интернет.....	107
ГЛАВА 14. Нахождение ошибок в готовых программах и их доработка	112
ГЛАВА 15. Написание простых программ	120
ГЛАВА 16. Нахождение выигрышной стратегии	127
ГЛАВА 17. Написание сложных программ.....	142
ГЛАВА 18. Вариант ЕГЭ повышенной сложности.....	148
ГЛАВА 19. Ответы и решения задач.....	158
Рекомендуемые интернет-ресурсы по ЕГЭ.....	181
Литература.....	183