

**ПОДГОТОВКА К ЕГЭ
ВЫСШИЙ УРОВЕНЬ КАЧЕСТВА**



2013

ИНФОРМАТИКА

ТЕМАТИЧЕСКИЕ ТРЕНИРОВОЧНЫЕ ЗАДАНИЯ

- ✓ Задания частей А, В и С по всем темам ЕГЭ
- ✓ Ответы ко всем заданиям



ЭКСМО



**ПОДГОТОВКА К ЕГЭ
ВЫСШИЙ УРОВЕНЬ КАЧЕСТВА**

ЕГЭ



2013

Н.Н. Самылкина, Е.М. Островская

ИНФОРМАТИКА

ТЕМАТИЧЕСКИЕ ТРЕНИРОВОЧНЫЕ ЗАДАНИЯ

Рекомендовано «Институтом содержания и методов обучения»
Российской академии образования



Москва **эксмо** 2012

УДК 373.167.1:004
ББК 32.81я7
С 17

Рецензент:

Ю. Г. Коротенков — кандидат физико-математических наук

Самылкина Н. Н.

С 17 ЕГЭ 2013. Информатика. Тематические тренировочные задания / Н. Н. Самылкина, Е. М. Островская. — М. : Эксмо, 2012. — 96 с. — (ЕГЭ. Тематические тренировочные задания).

ISBN 978-5-699-58095-8

Это издание адресовано *учащимся 10–11-х классов и абитуриентам*, сдающим экзамен по информатике в форме ЕГЭ. Тренировочные задания позволят систематически, при прохождении темы, готовиться к экзамену.

В рабочей тетради представлены:

- задания частей А, В, С по всем темам ЕГЭ;
- методические указания по выполнению заданий разного типа;
- ответы ко всем заданиям.

Книга будет полезна *учителям* информатики, так как дает возможность эффективно организовать подготовку учащихся к единому экзамену непосредственно на уроках, в процессе изучения всех тем.

УДК 373.167.1:004
ББК 32.81я7

ISBN 978-5-699-58095-8

© Самылкина Н. Н., Островская Е. М., 2012
© Оформление. ООО «Издательство «Эксмо», 2012

ВВЕДЕНИЕ

По сравнению с другими предметами, предмет «Информатика и ИКТ» имеет свои особенности, которые необходимо учитывать при подготовке к единому государственному экзамену. Большую часть учебного материала по предмету составляют практические умения в области ИКТ и программирования, которые затруднительно проверить в стандартном формате ЕГЭ. В то же время эти умения находятся в тесной связке с основным теоретическим содержанием предмета.

Поэтому перед разработчиками ЕГЭ по информатике и ИКТ стоит совсем непростая задача: в существующих условиях постараться проверить наиболее значимые элементы знаний и умений, необходимые для будущей профессиональной подготовки, при этом постараться не выйти за рамки стандарта общего образования по предмету. Все это говорит о том, что к экзамену по информатике и ИКТ необходимо специально уделить некоторое время на подготовку. Прежде всего просмотреть, какие темы проверяются и какими типами заданий это делается. Подготовка к ЕГЭ, как и подготовка к экзамену в традиционной форме, требует обобщающего повторения и систематизации имеющихся знаний. Наиболее успешный способ подготовки — подготовка с использованием тематических контрольных материалов и вариантов, аналогичных используемым на экзамене.

Как работать с изданием

Успешность сдачи единого государственного экзамена по информатике и ИКТ определяется прежде всего подготовкой выпускника по предмету и его уверенностью в успехе.

Предлагаем выпускникам проверить свои знания на материале пособия «ЕГЭ. Информатика. Тематические тренировочные задания», которое содержит четыре основных раздела курса информатики и ИКТ. По темам, входящим в каждый раздел, предлагаются типичные для вариантов ЕГЭ задания.

Раздел «Представление информации» представлен темами: «Кодирование информации», «Измерение информации», «Системы счисления», «Пропускная способность канала связи». Раздел «Основы логики» раскрывает последовательно именно эту тему. Раздел «Информационные и коммуникационные технологии» объединяет темы: «Программные средства информационных и коммуникационных технологий», «Технология обработки графической и звуковой ин-

формации», «Технология обработки информации в электронных таблицах», «Технология хранения, поиска и сортировки информации в базах данных», «Телекоммуникационные технологии» и «Моделирование и компьютерный эксперимент». Раздел «Алгоритмизация и программирование» содержит только эту тему, но она входит во все части экзаменационной работы заданиями разного уровня сложности. Поэтому этот раздел объединяет как самые простые задания на примере исполнителей, действующих в обстановке, так и самые сложные задачи на технологию программирования, находящиеся в части С.

После выполнения заданий можно сравнить свои результаты с эталоном и самостоятельно определить, какие из тем вызвали затруднения и требуют дополнительного повторения. Для повторения и тренировки по вариантам, аналогичным используемым на ЕГЭ, можно использовать издание *Самылкиной Н.Н., Островской Е.М.* «ЕГЭ. Информатика. Тренировочные задания».

Желаем удачи на экзамене!

Бланк ответов №2



Регион

Код предмета

Название предмета

Резерв - 8

Дополнительный бланк ответов №2

Лист №

1

Перепишите значение полей «регион», «код предмета», «название предмета» из БЛАНКА РЕГИСТРАЦИИ.
Отвечая на задание типа С, пишите аккуратно и разборчиво, соблюдая разметку страницы.
Не забудьте указать номер задания, на которое Вы отвечаете, например С1.
Условия задания переписывать не нужно.

ВНИМАНИЕ! Все бланки и листы с контрольными измерительными материалами рассматриваются в комплекте.

Large grid area for writing answers.

ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

► По разделу «Представление информации» предлагаются тренировочные задания по темам: «Кодирование информации», «Измерение информации», «Системы счисления», «Пропускная способность канала связи».

В первой части экзаменационной работы ЕГЭ располагаются задания с выбором одного правильного ответа из четырех предложенных. При решении заданий такого типа следует поставить знак «X» в клетке с номером правильного, по вашему мнению, ответа.

При выполнении заданий можно тренироваться в заполнении бланков, предлагаемых на экзамене (№ 1 и № 2). В специально отведенном месте, рядом с каждым заданием, расположен фрагмент бланка, который можно заполнять. По окончании работы следует перенести все ответы в образцы настоящих бланков.

Задания

1. Состояние морского аквариума контролируется сорока пятью датчиками. Наименьшее количество двоичных разрядов, необходимое для идентификации этих датчиков, равно ...

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

2. Выберите из перечисленных целое положительное число, которое можно записать в двухбайтовую ячейку памяти.

- 1) 256
2) 1024
3) 6750
4) 65 535

3. Цветной сканер имеет разрешение 600×600 точек/дюйм. Объем памяти, занимаемой просканированным изображением размером 2,5×4 дюйма, составляет около 10,8 Мбайт. Определите число возможных цветовых комбинаций сканера.

- 1) 256
2) 2^{16}
3) 2^{24}
4) 65 536

4. Цветной сканер имеет разрешение 400×400 точек/дюйм. Объем памяти, занимаемой просканированным изображением размером 2,5×4 дюйма, составляет около 4,8 Мбайт. Определите число возможных цветовых комбинаций сканера.

- 1) 65 536
2) 2^3
3) 3
4) 16 777 216

1 2 3 4 1

1 2 3 4 2

1 2 3 4 3

1 2 3 4 4

5 1 2 3 4

5. Монитор работает в режиме с разрешением 1920×1280 при глубине представления цвета 8 бит и частоте обновления экрана 75 Гц. Какую минимальную пропускную способность должен поддерживать видеоадаптер, работающий с монитором?

- 1) 176 Кбит/с
- 2) около 176 Мбайт/с
- 3) не менее 100 байт/с
- 4) 150 Мбайт/с

6 1 2 3 4

6. Для кодирования черно-белого полутонового изображения используются 256 уровней серого цвета. Какой объем памяти займет изображение размером 1000×3000 точек?

- 1) около 0,8 Мбайт
- 2) более 7,7 Мбайт
- 3) 5000 Кбайт
- 4) около 3 Мбайт

7 1 2 3 4

7. Четырехугольник в трехмерном пространстве задан координатами вершин. Сколько ячеек памяти компьютера потребуется для представления в ней такого четырехугольника, если одно число занимает одну ячейку памяти?

- 1) 16
- 2) 12
- 3) 8
- 4) 4

8 1 2 3 4

8. Латинские буквы А, В, С, D закодированы двоичными числами (000, 010, 011, 110 соответственно). Какая последовательность букв закодирована в следующей двоичной записи: 110000110011?

- 1) DCAD
- 2) DBAD
- 3) DACD
- 4) DADC

9 1 2 3 4

9. Буквы кириллицы закодированы следующим образом

А	Б	В	Г	Д	Е	Ж
101	110	111	1000	1001	1010	1011

Если таким образом закодировать последовательность символов ГДЕЖЕБЕГ и записать результат в восьмеричной системе счисления, то получится:

- 1) 1046535325
- 2) 4232565524
- 3) 10465353250
- 4) 42325655240

10. Сообщение на английском языке первоначально было записано в 16-битном коде Unicode. При его перекодировке в 8-битную кодировку КОИ-8 информационное сообщение уменьшилось на 960 бит. Сколько символов содержалось в сообщении?

- 1) 30
- 2) 60
- 3) 120
- 4) 480

1 2 3 4 10

11. Учитель информатики сообщил алгоритм получения пароля для входа в операционную систему. Пароль состоит из 6 символов. Из исходной записи «SCHOOL555INFORM» удалить повторяющиеся символы, а затем все символы, начиная с седьмого (т.е. оставить первые 6 символов). Какая из предложенных записей является паролем?

- 1) SCHLIN
- 2) SCH555
- 3) SCHOOL
- 4) SCHINF

1 2 3 4 11

12. Среди приведенных чисел выберите наибольшее.

- 1) 1111100_2
- 2) 315_8
- 3) 1200_4
- 4) $F3_{16}$

1 2 3 4 12

13. Среди приведенных чисел выберите наименьшее.

- 1) 1111101_2
- 2) 125_8
- 3) 2103_4
- 4) $4A_{16}$

1 2 3 4 13

14. Дано $a = 82_{16}$, $b = 701_8$. Выберите число c , записанное в двоичной системе, которое отвечает условию $a < c < b$.

- 1) 10001001
- 2) 10001100
- 3) 11010111
- 4) 1111111000

1 2 3 4 14

15. Вычислите сумму чисел 301_8 и 47_{16} .

- 1) 70_8
- 2) 170_8
- 3) 740_{16}
- 4) 1100001_2

1 2 3 4 15

Во второй части экзаменационной работы располагаются задания, ответ на которые необходимо получить самостоятельно и записать в специально отведенное для этого место.

При выполнении этих заданий также можно тренироваться в заполнении бланков, предлагаемых на экзамене (№ 1 и № 2). В специально отведенном месте, рядом с каждым заданием, расположен фрагмент бланка, который можно заполнять. По окончании работы следует перенести все ответы в образцы настоящих бланков.

16

16. Сколько единиц в двоичной записи шестнадцатеричного числа 83?

17

17. Укажите через запятую, в каких системах счисления (до десятичной) число 10 нечетное?

18

18. Состояние охраняемого объекта контролируют 76 датчиков. Определите наименьшее количество сигнальных лампочек, необходимых для идентификации этих датчиков и передачи с них информации.

19

19. Электронный почтовый ящик имеет объем 10 Мбайт. Информация на его адрес по открытому на прием каналу связи передается со скоростью 3,6 Кбайт/с. Через какое время у поставщика услуг электронной почты появится повод прислать извещение о переполнении почтового ящика при непрерывном приеме писем? Укажите время в минутах, округлив значение до целых.

20

20. Скорость передачи данных через ADSL-соединение равна 2,5 Мбайт/с. Через данное соединение передают файл размером 512 Кбайт. Определите время передачи файла в секундах. [7]

21

21. Символьная запись представляет собой набор из двоичных чисел и знаков «#». Каждая строка составлена по правилу: в начале строки знак «#», далее число в двоичном виде и знак «#».

Например, первые три строки выглядят так:

(1) #1#

(2) #10#

(3) #11#

Сколько единиц в записи 15-й строки перед знаком решетки?

22

22. Даны 9 строк, которые нумеруются от 0 до 8. Нулевая строка состоит из одного символа — 0 (ноль). Каждая из последующих строк создается такими действиями: в очередную строку дважды подряд приписывается предыдущая строка и в конце дважды приписывается номер строки.

Вот первые 4 строки, созданные по этому правилу:

- (0) 0
- (1) 0011
- (2) 0011001122
- (3) 00110011220011001122 33

Запишите через запятую два символа, стоящих в последней строке на **765** и **766**-м месте (считая слева направо), и количество нулей. [7]

- 23.** Строки (цепочки символов букв греческого алфавита) создаются по следующему правилу.

Первая строка состоит из одного символа — греческой буквы «Ω». Каждая из последующих строк создается следующим образом: в очередную строку сначала записывается дважды предыдущая строка, а в конец приписывается еще один символ, чей порядковый номер в греческом алфавите соответствует номеру строки минус 1.

Вот первые 5 строк, созданные по этому правилу:

- (1) Ω
- (2) ΩΩΑ
- (3) ΩΩΑ ΩΩΑ Β
- (4) ΩΩΑ ΩΩΑΒ ΩΩΑ ΩΩΑΒ Γ
- (5) ΩΩΑ ΩΩΑΒ ΩΩΑ ΩΩΑΒ Γ ΩΩΑ ΩΩΑΒ ΩΩΑ ΩΩΑΒ Γ Δ

Греческий алфавит (для справки):

Α Β Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω

Какой символ стоит в девятой строке на **505**-м месте (считая слева направо)?

ОСНОВЫ ЛОГИКИ

По разделу «Основы логики» предлагаются тренировочные задания только по этой теме, которыми проверяются знания и умения выпускников создавать и преобразовывать логические выражения в соответствии с законами логики, читать таблицу истинности, а также решать текстовые логические задачи любым удобным для этого способом (алгебраическим или составлением логического выражения).

В первой части экзаменационной работы ЕГЭ располагаются задания с выбором одного правильного ответа из четырех предложенных. При решении заданий такого типа следует поставить знак «х» в клетке с номером правильного, по вашему мнению, ответа.

При выполнении заданий можно тренироваться в заполнении бланков, предлагаемых на экзамене (№ 1 и № 2). В специально отведенном месте, рядом с каждым заданием, расположен фрагмент бланка, который можно заполнять. По окончании работы следует перенести все ответы в образцы настоящих бланков.

Задание

1 1 2 3 4

1. Формулой логического высказывания

«В июле мы с друзьями поедem на Селигер, и если катер будет на ходу, то рыбалка должна быть удачной»
является...

- 1) $A \wedge (B \rightarrow C)$
- 2) $(A \wedge B) \vee C \vee D$
- 3) $(A \wedge B) \leftrightarrow (C \wedge D)$
- 4) $A \wedge B \rightarrow C$

2 1 2 3 4

2. Какое логическое выражение равносильно данному:

$$X \wedge Y \vee X \wedge \neg Y?$$

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

3 1 2 3 4

3. Дано логическое выражение: $Y \wedge \neg X \wedge (Y \rightarrow X)$. Определите его значение.

- 1) 1
- 2) 0
- 3) X
- 4) Y

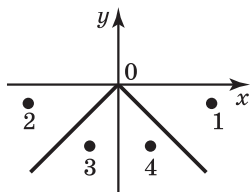
4. Какое логическое выражение равносильно данному:

$$A \wedge B \vee \neg A \wedge C \vee B \wedge C?$$

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

1 2 3 4 4

5. На плоскости задан график $y = -x$.



1 2 3 4 5

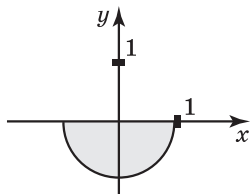
Укажите номер точки, координаты которой удовлетворяют условию:

$$(y > x) \wedge (y < 0) \wedge (x > 0)$$

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

6. Укажите логическое выражение, значение которого истинно, если точка с координатами x и y попадает в заштрихованную область — часть круга с центром в начале координат и радиусом, равным 1 — ...

1 2 3 4 6



- 1) $((x < 1) \wedge (x > -1)) \vee ((y < 1) \wedge (y > 0))$
- 2) $((x^2 + y^2) < 1) \wedge (y < 0)$
- 3) $(x < 1) \wedge (x > -1) \wedge (y < 1) \wedge (y > 0)$
- 4) $((x^2 + y^2) < 1) \wedge \neg((y > 0) \wedge (x > 0))$

7. Дана таблица истинности некоторой логической функции F от трех аргументов: X, Y, Z.

1 2 3 4 7

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

8

Из перечисленных выражений выберите F.

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

8. Дана таблица истинности некоторой логической функции F от трех аргументов: A, B, C.

A	B	C	F
0	1	1	0
1	0	0	1
1	0	1	0

Из перечисленных выражений выберите F.

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

► Во второй части экзаменационной работы располагаются задания, ответ на которые необходимо получить самостоятельно и записать в специально отведенное для этого место.

При выполнении этих заданий также можно тренироваться в заполнении бланков, предлагаемых на экзамене (№ 1 и № 2). В специально отведенном месте, рядом с каждым заданием, расположен фрагмент бланка, который можно заполнять. По окончании работы следует перенести все ответы в образцы настоящих бланков.

9

9. Дано тождественно-истинное логическое выражение:

$$(A \wedge \neg B) \wedge (B \vee \neg C \wedge D) = 1,$$

где A, B, C, D — логические переменные. Определите, сколько различных решений оно имеет?

В ответе не перечисляйте все различные наборы значений A, B, C и D, при которых выполняется данное равенство. В качестве ответа нужно указать только количество таких наборов. [6]

10

10. Сколько различных решений имеет уравнение $(P \wedge Q \vee \neg S) = 1$?

P, Q, S — логические переменные. В ответе **не перечисляйте** все наборы значений, при которых выполняется равенство, а укажите количество таких наборов.

11

11. Определите наибольшее целое число X, при котором истинно логическое высказывание $(X * X - 45 > 0) \wedge (100 > (X + 10) * (X - 5))$.

12. В магазине продаются большие и маленькие речные раки. Большой рак стоит в два раза дороже маленького. Александр Петрович купил 5 больших раков и 3 маленьких. Если бы он купил 3 больших рака и двух маленьких, то потратил бы на 20 рублей меньше. Сколько стоит маленький рак?

12

13. Однажды в лесу звери устроили соревнование по бегу с препятствиями. В соревновании участвовали: Белка (Б), Заяц (З), Обезьяна (О) и Лисица (Л). По окончании болельщики затеяли спор о том, кто за кем пришел к финишу. Звери утверждали:

13

- 1) Белка (Б) пришла второй, а Заяц (З) был третьим.
- 2) Белка (Б) была первой, а Лисица (Л) пришла второй.
- 3) Обезьяна (О) была второй, а Заяц (З) пришел четвертым.

На самом деле каждый из зверей ошибся один раз.

В каком порядке участники соревнования пришли к финишу?

В ответе укажите первые буквы имен.

14. На очередном конкурсе «Евровидение» зрители поспорили, участник какой страны будет победителем. Были высказаны следующие предположения:

14

— первым будет исполнитель из России, а вторым участник из Белоруссии;

— исполнитель из России будет вторым, а из Норвегии — третьим;

— вторым будет участник из Франции, а норвежский исполнитель будет четвертым.

Оказалось, в каждом из высказанных предположений одно — истинно, другое — ложно. Определите победителей, написав строку из 4 символов — первых букв названий стран, получивших места с первого по четвертое (в порядке занятых мест). (Буквы заглавные).

Например, ответ мог бы выглядеть так: БРФН.

15. Андрей (А), Федор (Ф), Валерий (В) и Григорий (Г) сопровождали своих сестер на бал. Заключительный танец каждая из девушек танцевала не со своим братом. Образовались следующие пары: Лена (1) с Андреем, Анна (2) с братом Кати (З), Тома (4) с братом Анны, Федор с сестрой Валерия, а Валерий с сестрой Андрея. Определите, кому приходится сестрой каждая из девочек? Выпишите пары: брат—сестра, проставив первую букву имени мальчика и соответствующий номер девочки. Например, ответ мог бы выглядеть так: А1, Г2 и т.д. [3]

15

16. В Солнечном городе на одной улице живут 4 друга Незнайки: Винтик, Шпунтик, Торопышка и Коротышка. Известно, что каждый из них вырос и получил профессию (пекарь, слесарь, водитель, портной), по которой и работает. Но Незнайка забыл,

16

кто из друзей по какой профессии работает и где живет. Однако ему удалось вспомнить, что:

- 1) Пекарь живет левее слесаря.
- 2) Водитель живет правее портного.
- 3) Портной живет рядом со слесарем.
- 4) Пекарь живет не рядом со слесарем.
- 5) Торопышка живет правее портного.
- 6) Коротышка не пекарь.
- 7) Шпунтик живет рядом со слесарем.
- 8) Торопышка живет левее Шпунтика.

Помогите Незнайке определить, кто из его друзей где живет, и дайте ответ в виде заглавных букв их имен (слева направо). Например, если бы в домах жили (слева направо) Торопышка, Винтик, Шпунтик и Коротышка, ответ был бы: ТВШК [7].

В третьей части экзаменационной работы располагаются самые сложные и трудоемкие задания, на которые необходимо представить развернутое решение или обоснование полученного ответа. Полное решение и полученный ответ записывается четко и разборчиво в бланк ответов № 2.

17

17. Двое играют в следующую игру. Из имеющейся кучки из 11 камней по очереди берут камни в количестве 1, 2 или 3 по своему усмотрению. Выигрывает тот, кто возьмет последний камень. Опишите выигрышную стратегию игры. [3]

18

18. **Задание из демоверсии 2010 года.** 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. [7]

19

19. Два игрока придумали игру в фишки на координатной плоскости. 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 $(x + 3, y)$, или в точку с координатами $(x, y + 2)$, или в точку с координатами $(x, y + 4)$. Выигрывает игрок, после хода которого расстояние

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. [1]

20. Два игрока играют в фишки на координатной плоскости. В начале игры фишка находится в точке с координатами $(7, 15)$. Игроки ходят по очереди. Ход состоит в том, чтобы переместить фишку из точки с координатами (x, y) в одну из трех точек с координатами: $(x - 2, y)$, $(x, y - 2)$, $(x, y - 3)$. Выигрывает игрок, после хода которого расстояние от фишки до точки с координатами $(0, 0)$ не более 11 единиц.

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

ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ. МОДЕЛИРОВАНИЕ И КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

► По разделу «Информационные и коммуникационные технологии» и «Моделирование и компьютерный эксперимент» предлагаются тренировочные задания по темам: «Программные средства информационных и коммуникационных технологий», «Технология обработки графической и звуковой информации», «Технология обработки информации в электронных таблицах», «Технология хранения, поиска и сортировки информации в базах данных», «Телекоммуникационные технологии» и «Представление и считывание данных в разных типах информационных моделей (схемы, карты, таблицы, графики, формулы)».

В первой части экзаменационной работы ЕГЭ располагаются задания с выбором одного правильного ответа из четырех предложенных. При решении заданий такого типа следует поставить знак «х» в клетке с номером правильного, по вашему мнению, ответа.

При выполнении заданий можно тренироваться в заполнении бланков, предлагаемых на экзамене (№ 1 и № 2). В специально отведенном месте, рядом с каждым заданием, расположен фрагмент бланка, который можно заполнять. По окончании работы следует перенести все ответы в образцы настоящих бланков.

Задания

1. Дан фрагмент электронной таблицы:

	A	B	C	D
1				
2				
3		Классный журнал		
4				

Укажите адрес ячейки, в которой содержится текст: «Классный журнал».

- 1) C3
- 2) B2
- 3) D3
- 4) B3

2. Сколько ячеек в блоке С4 : Е7 электронной таблицы?

- 1) 13
- 2) 12
- 3) 9
- 4) 6

1 2 3 4 2

3. В ячейку А1 электронной таблицы EXCEL введено слово «январь».

	А	В	С	Д
1	январь	→		
2				
3				
4				

1 2 3 4 3

Ячейка А1 выделяется, и маркер автозаполнения + «протягивается» вправо. Что появится в ячейке В1?

- 1) курсор
- 2) «февраль»
- 3) «январь»
- 4) «декабрь»

4. В ячейки А1 и А2 электронной таблицы EXCEL введены числа 11 и 12. Затем ячейки А1 и А2 выделяют и маркер заполнения + «протягивают» вниз.

Что появится в ячейке А3?

	А	В	С	Д
1	11			
2	12			
3	↓			

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

1 2 3 4 4

5. В электронной таблице в блок А1:А3 введены последовательно числа — 5, 6, 7, а в блок В1:В3 — 10, 12, 14.

Значение ячейки С1 вычисляется по формуле: = А1 + В\$1 - А\$2.

	А	В	С	Д
1	5	10	9	
2	6	12	↓	
3	7	14		

1 2 3 4 5

Какое число появится в ячейке С2, если выделить ячейку С1 и маркер заполнения + «протянуть» вниз?

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

6 1 2 3 4

6. В рабочей книге электронной таблицы EXCEL заданы два листа с именами «Количество» и «Сумма». Для вычисления значения ячейки А1 листа «Сумма» используется формула, в которой число 30 умножается на значение ячейки В2 листа «Количество».

	А	В	С
1			
2		100	
3			
4			

Количество

	А	В	С
1			
2			
3			
4			

Сумма

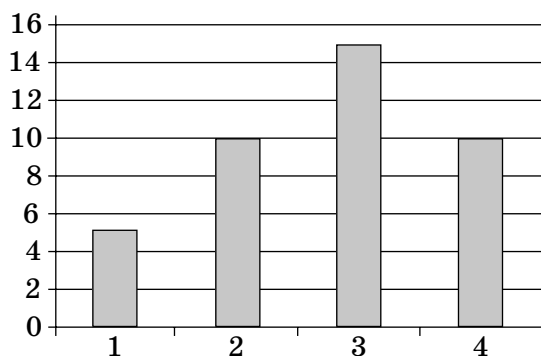
Укажите необходимую формулу.

- 1) = 30 * Количество!В2
- 2) = 30* В2
- 3) = 30 * 100
- 4) = 30 * [Количество]В2

7 1 2 3 4

7. По данным электронной таблицы EXCEL построена диаграмма.

	А	В	С	Д
1	5	5	20	5
2	15	10	10	5
3	10	15	0	5
4	15	10	10	5



Укажите имя столбца таблицы, данные которого отражены на диаграмме.

- 1) А
- 2) В
- 3) С
- 4) D

8. Задан фрагмент таблицы базы данных.

1 2 3 4 8

AddressID	FirstName	LastName	Adress	HomesPhone
1	Петренко	Андрей	ул. Маршала Полубоярова, 45, 5	735-50-55
2	Скрупакова	Анна	ул. Андрея Сахарова, 25, 5	659-51-55
3	Кошелева	Ольга	ул. Кирова, 10, 2	303-32-32
4	Зорина	Тамара	ул. Транспортная, 1, 14	727-79-77

Сколько записей будет найдено, если выбраны следующие условия поиска:

поле поиска — «**FirstName**»;

образец поиска — сочетание символов «ко»;

совпадение с образцом — в любой части поля?

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

9. Дан фрагмент таблицы базы данных.

1 2 3 4 9

AddressID	FirstName	LastName	Adress	HomesPhone
1	Дуров	Валентин	Проспект Мира, 8, 20	184-20-83
2	Ковалев	Евгений	ул. Жулебинская, 6, 48	521-12-40
3	Планкова	Евгения	ул. Киевская, 5, 25	243-12-17
4	Ковалевский	Виктор	Жулебинский пр-кт, 28, 5	243-20-27
5	Дутова	Валентина	Кутузовский проезд, 4, 6	243-67-07

По данным таблицы требуется восстановить номер телефона абонента, о котором известно, что его фамилия либо Ковалев, либо Ковалевский, что проживает он на Жулебинском проспекте и что номер его телефона оканчивается на цифру 7. Чтобы найти номер телефона этого абонента, необходимо применить следующий фильтр ...

- 1) (**FirstName**= «К*») И (**Adress**= «Ж*») И (**HomesPhone**=#7)
- 2) (**FirstName**= «Ковал*») И (**Adress**= «Жулебин*») И (**HomesPhone**= ### - ## -#7)
- 3) (**FirstName**= «Ковалев*») И (**Adress**= «Жулебинский*») И (**HomesPhone**=### - ## - #7)
- 4) (**FirstName**= «К*») И (**Adress**= «Жулебинский проспект*») И (**HomesPhone**=#7)

10. Задана таблица базы данных.

Класс	Фамилия	Год	Месяц	Рост
10 А	Афанасьев	1988	1	162
10 Б	Валовой	1987	3	158
10 А	Донцова	1986	2	155
10 Б	Александрова	1989	4	151

Укажите фамилии учащихся, данные о которых появятся в таблице запроса, если выбрано следующее условие отбора:

(Класс = «10 А») ИЛИ (Класс = «10 Б») И (Год < 1988) И (Месяц <= 3).

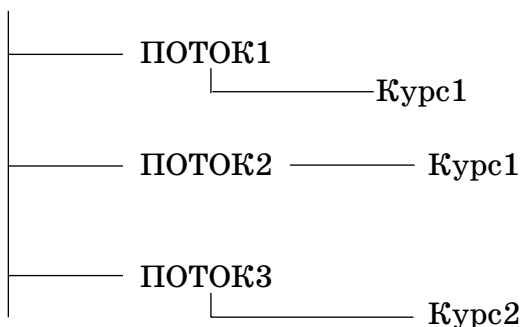
- 1) Афанасьев и Александрова
- 2) Донцова и Валовой
- 3) Валовой и Александрова
- 4) Афанасьев и Донцова

11. Файл `pusk.exe` записан на диске D, в каталоге BASIC, подкаталоге BIN. Выберите правильный путь доступа к файлу.

- 1) D:\BASIC\BIN\pusk.exe
- 2) pusk.exe
- 3) D:\BIN\ pusk.exe
- 4) D:\BASIC\pusk.exe

12. Дано дерево каталогов:

A:/



Определите полное имя файла Курс2.

- 1) A:/ПОТОК1/Курс2
- 2) A:/ПОТОК2/Курс1/Курс2
- 3) A:/ПОТОК1/ПОТОК3/Курс2
- 4) A:/ПОТОК3/Курс2

13. Файл, полное имя которого было D:\Мои_документы\KLASS\spisok1, сохранили в подкаталоге ARHIV корневого каталога диска K. Определите полное имя сохраненного файла.

- 1) K:\Мои_документы\ARHIV\spisok1
- 2) K:\KLASS\ARHIV\spisok1
- 3) K:\ARHIV\spisok1
- 4) K:\spisok1

1 2 3 4 13

14. Для выполнения операций с несколькими файлами одновременно используются **маски имен файлов**. Маска может содержать последовательность букв, цифр и прочих допустимых в именах файлов символов. Чаще всего используют следующие символы:

Символ «?» (вопросительный знак) означает один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может обозначать и пустую последовательность.

Выберите имя файла, удовлетворяющее маске: v?s*.doc.

- 1) vedomosti.doc
- 2) vesti.doc
- 3) vird.doc
- 4) vse_documenti.doc

1 2 3 4 14

15. Выберите из перечисленных IP-адрес конкретный узел.

- 1) 197.96.557.14
- 2) 224.16.81.8
- 3) 81.19.85.116
- 4) 0.0.81.18

1 2 3 4 15

16. Компьютер находится в подсети 172.16.16.0. Укажите IP-адрес, к которому он без посредников **не может** обратиться.

- 1) 172.16.16.12
- 2) 172.16.16.3
- 3) 172.16.16.1
- 4) 192.168.16.21

1 2 3 4 16

17. Укажите протокол, **не позволяющий** передать видеоролик.

- 1) HTTP
- 2) SMTP
- 3) DNS
- 4) FTP

1 2 3 4 17

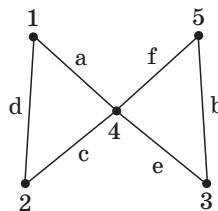
18. В таблице приведены итоги шахматного турнира:

Участники	Антонов	Беляев	Валиев	Григорьев
Антонов	0	1/2	1	1/2
Беляев	1/2	0	1/2	0
Валиев	0	1/2	0	1
Григорьев	1/2	1	0	0

Кто является победителем турнира?

- 1) Антонов
- 2) Беляев
- 3) Валиев
- 4) Григорьев

19. Дан граф:



Выберите его табличное представление.

1)

	1	2	3	4	5
1		a		b	
2	a			c	
3				f	d
4	b	c	f		e
5			d	e	

2)

	1	2	3	4	5
1		a		b	
2		a		c	
3				f	d
4	b	c	f		e
5			d	e	

3)

	1	2	3	4	5
1		d		a	
2	d			c	
3				e	b
4	a	c	e		f
5			b	f	

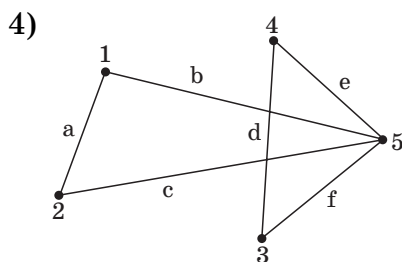
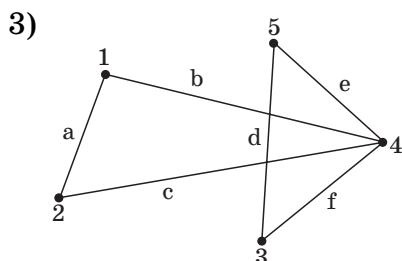
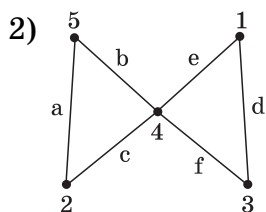
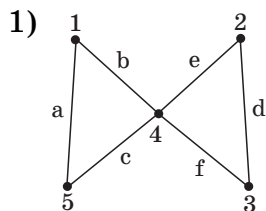
4)

	1	2	3	4	5
1	a	a		b	
2				c	
3				f	d
4	b	c	f	d	e
5				e	

20. Дано следующее описание графа:

$a(2, 5)$ $b(4, 5)$ $c(2, 4)$ $d(1, 3)$ $e(1, 4)$ $f(3, 4)$

Укажите его графическое представление



► Во второй части экзаменационной работы располагаются задания, ответ на которые необходимо получить самостоятельно и записать в специально отведенное для этого место.



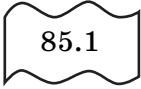
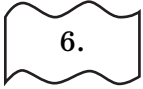
При выполнении этих заданий также можно тренироваться в заполнении бланков, предлагаемых на экзамене (№ 1 и № 2). В специально отведенном месте, рядом с каждым заданием, расположен фрагмент бланка, который можно заполнять. По окончании работы следует перенести все ответы в образцы настоящих бланков.

21. Доступ к файлу `tryam.avi`, находящемуся на сервере `www.net`, осуществляется по протоколу `ftp`. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла. [7]

А	<code>://</code>	Д	<code>.avi</code>
Б	<code>tryam</code>	Е	<code>/</code>
В	<code>ftp</code>	Ж	<code>www</code>
Г	<code>.net</code>		

22

22. Расставьте фрагменты IP-адреса в верной последовательности.

			
А	Б	В	Г

23

23. В языке запросов поисковой системы знак «!» перед словом означает, что слово должно быть использовано именно в таком виде. В таблице приведены поисковые запросы, одновременно выданные к одной поисковой системе. Расположите их в порядке возрастания количества страниц, соответствующих запросу.

- 1) Рикша, истребитель или трамвай до !Дели
- 2) Рикша, истребитель или трамвай до Дели
- 3) (Рикша | истребитель | трамвай)&(Дели)
- 4) Рикша & истребитель & трамвай & Дели

24

24. Абонент скачивает из файлообменной сети фильм объемом 385 Мбайт. Фильм скачивается из пяти разных источников, каждый из которых отдает в среднем 35 Кбайт в секунду. Сколько времени займет скачивание фильма? В ответе укажите только число в минутах, округлив его до целых. Слово «минут» или «мин» добавлять не нужно.

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

► По разделу «Алгоритмизация и программирование» предлагаются тренировочные задания по темам: «Алгоритмизация и основы программирования» и «Технология программирования», включенные во все три части экзаменационной работы.

В первой части экзаменационной работы ЕГЭ располагаются задания с выбором одного правильного ответа из четырех предложенных. При решении заданий такого типа следует поставить знак «х» в клетке с номером правильного, по вашему мнению, ответа.

При выполнении заданий можно тренироваться в заполнении бланков, предлагаемых на экзамене (№ 1 и № 2). В специально отведенном месте, рядом с каждым заданием, расположен фрагмент бланка, который можно заполнять. По окончании работы следует перенести все ответы в образцы настоящих бланков.

Задания

1. Чему будет равно значение переменной *s* после выполнения фрагмента программы?

1 2 3 4 1

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
10 X = 0	x := 0;	x := 0
20 Y = X	y := x;	y := x
30 H1 = 1	h1 := 1;	h1 := 1
40 H2 = 2	h2 := 2;	h2 := 2
50 S = 0	s := 0;	s := 0
60 WHILE (X<=2) and (Y<=5)	While (x<=2) and (y<=5) do	<u>пока</u> (x<=2) <u>и</u> (y<=5)
70 S = S + 2	begin	s := s+2
80 X = X + H1	s := s + 2;	x := x + h1
90 Y = Y + H2	x := x + h1;	y := y + h2
100 WEND	y := y + h2;	кц
	end;	

- 1) 0
- 2) 2
- 3) 6
- 4) 4

2. Чему будет равно значение переменной **z** после выполнения фрагмента программы?

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
<pre>X=-3 CALL DIF(-3,Z) z = x - 2*z PRINT Z END SUB DIF(X,Z) Y = X + 5 x=x+2 IF (Y>0) AND (Y<5) THEN Z = X * Y ELSE Z = X + Y END IF END SUB</pre>	<pre>Var x,z:integer; procedure dif(x:integer;var z:integer); var y:integer; begin y := x+5; x:=x+2; if (y>0) and (y<5) then z := x * y else z := x + y; end; begin x := -3; dif(x,z); z := x - 2*z; writeln(z); end.</pre>	<pre>Алг dif(арг цел x,рез цел z) у := x+5 x:=x+2 если (y>0) и (y<5) то z := x * y иначе z := x + y все конец алг начало основной про- граммы x := -3 вызов dif(-3,z) z := x - 2*z Вывод z конец</pre>

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

3. Для определения наименьшего из трех значений переменных **a**, **b** и **c** необходимо написать фрагмент программы...

	БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
1)	<pre>10 IF (A<B) OR (A<C) THEN D = A 20 PRINT D</pre>	<pre>if (a<b) or (a<c) then d :=a; WriteLn (d);</pre>	<pre>если (a<b) или (a<c) то d := a все напечатать d</pre>
2)	<pre>10 D = A 20 IF B < D THEN D = B 30 IF C < D THEN D = C 40 PRINT D</pre>	<pre>d := a; if b < d then d := b; if c < d then d := c; WriteLn (d);</pre>	<pre>d := a если b < d то d := b все если c < d то d := c все напечатать d</pre>
3)	<pre>10 IF B<A<C THEN D = C 20 PRINT D</pre>	<pre>if (b<a<c) then d :=c; WriteLn (d);</pre>	<pre>если (b<a<c) то d := c все напечатать d</pre>
4)	<pre>10 IF C<A<B THEN D = B 20 PRINT D</pre>	<pre>if (c<a<b) then d :=b; WriteLn (d);</pre>	<pre>если (c<a<b) то d := b все напечатать d</pre>

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

4. Чему будет равно значение переменной **a** после выполнения фрагмента программы?

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
<pre> 10 X= 10 20 H1 = -2 30 H2 = 2: H3=3 40 A = -1 50 WHILE (X>=3) 60 Y = 2 70 WHILE (Y<=3) 80 A = A + h3 90 Y = Y + H2 100 WEND 110 X = X + H1 120 WEND </pre>	<pre> x := 10; h1 := -2; h3:=3; h2 := 2; a := -1; While (x >=3) do begin y := 2; While (y <= 3) do begin a := a + h3; y := y + h2 end; x := x + h1; end; </pre>	<pre> x := 10 h1 := -2 h2 := 2 h3:=3 a := -1 пока (x >= 3) y := 2 пока (y <= 3) a := a + 1 y := y + h2 кц x := x + h1 кц </pre>

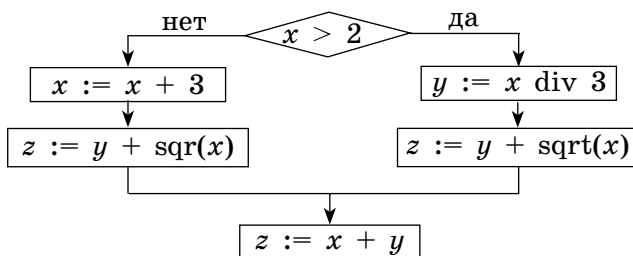
- 1) 11
- 2) 2
- 3) 5
- 4) 4

5. Чему будет равно значение переменной **b** после выполнения фрагмента программы?

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
<pre> 10 B= 0 20 A = -1 30 FOR I = 3 TO 5 40 X = A * I 50 A = - A 60 B = B + X 70 NEXT I </pre>	<pre> b := 0; a := -1; for i :=3 to 5 do begin x := a * i; a := -a; b := b + x; end; </pre>	<pre> b := 0 a := -1 НЦ ДЛЯ i ОТ 3 ДО 5 x := a * i a := -a b := b + x кц </pre>

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

6. Задан фрагмент блок-схемы алгоритма



где $x \text{ div } y$ — операция div — деление нацело x на y ;
 $\text{Sqr}(x)$ — функция возведения в квадрат аргумента x , т.е. x^2 ;
 $\text{Sqrt}(x)$ — функция извлечения квадратного корня из x .

Ему соответствует фрагмент программы ...

	БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
1)	<pre>IF X>2 THEN Y = X + 3 Z = Y + X^2 Else Y = X div 3 Z = Y + sqr(X) End if Z = X + Y</pre>	<pre>if x > 2 then begin y := x + 3; z := y + sqr(x); end else begin y := x div 3; z := y + sqrt(x); end; z := x + y;</pre>	<pre><u>если</u> x > 2 <u>то</u> y := x + 3 z := y + sqr(x) <u>иначе</u> y := x div 3 z := y + sqrt(x) <u>все</u> z := x + y</pre>
2)	<pre>IF X>2 THEN Y = X \ 3 Z = Y + X^2 Else Y = X + 3 Z = Y + sqr(X) End if Z = X + Y</pre>	<pre>if x > 2 then begin y := x div 3; z := y + sqr(x); end else begin y := x + 3; z := y + sqrt(x); end; z := x + y;</pre>	<pre><u>если</u> x > 2 <u>то</u> y := x \ 3 z := y + sqr(x) <u>иначе</u> y := x + 3 z := y + sqrt(x) <u>все</u> z := x + y</pre>
3)	<pre>IF X>2 THEN Y = X \ 3 Z = Y + sqr(X) Else Y = X + 3 Z = Y + X^2 End if Z = X + Y</pre>	<pre>if x > 2 then begin y := x div 3; z := y + sqrt(x); end else begin y := x + 3; z := y + sqrt(x); end; z := x + y;</pre>	<pre><u>если</u> x > 2 <u>то</u> y := x \ 3 z := y + sqrt(x) <u>иначе</u> y := x + 3 z := y + sqrt(x) <u>все</u> z := x + y</pre>
4)	<pre>IF X<2 THEN Y = X / 3 Z = Y + sqr(X) Else Y = X + 3 Z = Y + X^2 End if Z = X + Y</pre>	<pre>if x < 2 then begin y := x / 3; z := y + sqrt(x); end else begin y := x + 3; z := y + sqrt(x); end; z := x + y;</pre>	<pre><u>если</u> x < 2 <u>то</u> y := x / 3 z := y + sqrt(x) <u>иначе</u> y := x + 3 z := y + sqrt(x) <u>все</u> z := x + y</pre>

1) 1

2) 2

3) 3

4) 4

7 1 2 3 4

7. Задан фрагмент программы, обрабатывающий массив X размером N. После выполнения фрагмента программы

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
<pre>Imax:=1;imin:=1 For i:=2 to n If X(i)<x(imin) then imin=i else If X(i)>x(imax) then imax=i End if End if Next i y= X(imax) x(imax)=x(imin) x(imin)=y</pre>	<pre>Imax:=1;imin:=1; For i:=2 to n do Begin If X[i]<x[imin] then imin=i else If X[i]>x[imax] then imax=i; End; y:= X[imax]; x[imin]:=x[imax]; x[imax]:=y;</pre>	<pre>Imax:=1 imin:=1 <u>Нц для I от 2 до n</u> <u>если</u> X[i]<x[imin] <u>то</u> imin:=I <u>иначе</u> <u>если</u> X[i]>x[imax] <u>то</u> imax:=I <u>все</u> <u>все</u> <u>кц</u> y:= X[imax] x[imax]:=x[imin] x[imin]:=y</pre>

- 1) массив остался без изменений
- 2) максимальный и минимальный элементы поменялись местами
- 3) максимальный элемент стал первым в массиве
- 4) минимальный элемент стал первым в массиве

8. Дан фрагмент программы

1 2 3 4 8

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
Const n=5 Const m=4 Defint I,j Dim A(n,m) as integer Begin For i=1 to n do For j=1 to m do If i<j then A(I,j)=1 else A(I,j)=2 End if Next j Next i ...	Const n=5; Const m=4; Var I,j:integer; A: array[1..n,1..m] of integer; Begin For i:=1 to n do For j:=1 to m do If i<j then A[i,j]:=1 else A[i,j]:=2; ...	Константа n=5 Константа m=4 Арг цел I,j Арг цел таб A[1:n,1:m] <u>начало</u> <u>Нц для I от 1 до n</u> <u>Нц для j от 1 до m</u> <u>если i<j то</u> A[I,j]:=1 <u>иначе</u> A[I,j]:=2 <u>Все</u> <u>Кц</u> <u>кц</u> ...

Чему равна сумма значений элементов 1-й и 2-й строк массива A?

- 1) 5
- 2) 6
- 3) 8
- 4) 11

9. Имеется таблица, состоящая из пяти строк и пяти столбцов. Чему будет равна сумма элементов таблицы, стоящих под главной диагональю, после выполнения фрагмента программы?

1 2 3 4 9

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
10 FOR I = 1 TO 5 20 FOR J = 1 TO 5 30 IF J > I THEN 60 40 A(I, J) = I - J 50 GOTO 70 60 A(I, J) = I + J 70 NEXT J 80 NEXT I	for i := 1 to 5 do for j :=1 to 5 do if j > i then A [i, j] := i + j else A [i, j] := i - j ; ;	<u>нц для i от 1 до 5</u> <u>нц для j от 1 до 5</u> <u>если j > i</u> <u>то A[i, j] := i + j</u> <u>иначе A[i, j] := i - j</u> <u>все</u> <u>кц</u> <u>кц</u>

- 1) 10
- 2) 20
- 3) 30
- 4) 40

10. Задан массив X размером N. После выполнения фрагмента программы

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
<pre>Y=-1 For i=n to 1 step -1 X(i)=y*x(i) Y=-y Next i</pre>	<pre>Y:=-1; For i:=n downto 1 do begin X[i]:=y*x[i]; Y:=-y; End;</pre>	<pre>Y:=-1 Нц для I от n до 1 шаг -1 X[i]:=y*x[i] Y:=-y кц</pre>

- 1) массив остался без изменений
- 2) все элементы получили отрицательные значения
- 3) не менее половины элементов массива поменяли свой знак на противоположный
- 4) элементы массива с четными номерами поменяли свой знак на противоположный

11. Задан массив X размером N. После выполнения фрагмента программы

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
<pre>Y=x(1) For i=2 to n X(i-1)=x(i) Next i X(n)=y</pre>	<pre>Y:=x[1]; For i:=2 to n do X[i-1]:=x[i]; X[n]:=y;</pre>	<pre>Y:=x[1] Нц для I от 2 до n X[i-1]:=x[i] кц X[n]:=y</pre>

- 1) массив остался без изменений
- 2) все элементы, кроме первого, получили значение X[1]
- 3) первый элемент стал последним, а остальные переместились на позицию влево
- 4) последний элемент стал первым, а остальные переместились на позицию вправо

12. Задан фрагмент программы, обрабатывающий массив X размером N. После выполнения фрагмента программы

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
<pre>For i=1 to n-1 m=x(i) Im=I For j=i+1 to n If x(j)<m then m=x(j) im=j end if next j x(im)=x(i) x(i)=m Next i</pre>	<pre>For i:=1 to n-1 do begin m:=x[i]; Im:=I; For j:=i+1 to n do If X[j]<m then begin m:=x[j]; im:=j; end; x[im]:=x[i]; x[i]:=m; End;</pre>	<pre>Нц для I от 2 до n m:=x[i] Im:=I Нц для j от i+1 до n если X[j]<m то m:=x[j] im:=j все кц x[im]:=x[i] x[i]:=m кц</pre>

- 1) массив остался без изменений
- 2) массив упорядочен по убыванию
- 3) массив упорядочен по возрастанию
- 4) в массиве переставлены соседние пары элементов

13. Цепочка из 5 бусин, каждая из которых помечена набором символов: A1, A2, B1, B2 и C1, образована по следующему правилу. На первом месте находится бусина, не содержащая цифры 1. На последнем месте находится одна из бусин, не содержащая буквы В. В середине находится одна из бусин A2, B1, C1, которой нет на четвертом месте. На втором месте может быть одна из бусин A1, B1, C1, которой нет на третьем месте. Какая из перечисленных ниже цепочек создана по этому правилу?

- 1) C1A1A2B1B2
- 2) B2B1A1A1A2
- 3) A2C1B1C1A2
- 4) B1A1A2B2C1

14. Цепочка из 4 бусин, каждая из которых помечена одним из символов: A, B, C, D, E, F, образована по следующему правилу. На первом месте находится бусина, содержащая гласную. На втором месте находится одна из бусин B, D или гласная, которой нет на первом месте. На третьем месте находится одна из бусин B, C, D, F, не стоящая на 2-м месте. На последнем месте находится одна из бусин A, B, D, E, не стоящая на 2-м месте. Какая из перечисленных ниже цепочек создана по этому правилу?

- 1) ADCD
- 2) ADCE
- 3) EDBC
- 4) EDAC

15. Цепочка из 4 бусин, каждая из которых помечена набором символов: AA, BA, CA, AD и DA, образована по следующему правилу. На первом месте может находиться одна из бусин: AA, AD, DA. На последнем месте находится одна из бусин AD, DA, BA, не стоящая на 2-м месте. На втором месте находится одна из бусин DA, BA, CA, которой нет на 3-м месте. На 3-м месте может быть одна из бусин AA, AD, CA, которой нет на первом месте. Какая из перечисленных ниже цепочек создана по этому правилу?

- 1) AADAADBA
- 2) ADCACABA
- 3) AACAADCA
- 4) AABAADBA

1 2 3 4 13

1 2 3 4 14

1 2 3 4 15

16 1 2 3 4

16. Даны 4 числа: $A = 17$, $B = 81$, $C = 30$, $D = 64$. Были выполнены следующие действия: сравнили A и B , запомнили меньшее число как E . Затем сравнили C и D , запомнили большее число. И наконец, сравнили полученное число и число E и запомнили меньшее из них. Какое число запомнили?

- 1) 81
- 2) 64
- 3) 30
- 4) 17

17 1 2 3 4

17. Даны 4 числа: $A_1 = 15$, $A_2 = 3$, $A_3 = 40$, $A_4 = 7$. Были выполнены следующие действия: сравнили A_1 и A_2 , запомнили меньшее число как B_1 . Затем сравнили A_3 и A_4 , запомнили меньшее число как B_2 . Сравнили B_1 и B_2 , запомнили меньшее из них. Какое число запомнили?

- 1) 40
- 2) 15
- 3) 7
- 4) 3

18 1 2 3 4

18. Даны 3 числа: $A = 60$, $B = 70$, $C = 45$. Были выполнены следующие действия: сравнили A и B , запомнили большее число. Затем сравнили полученное число и удвоенное число C и запомнили меньшее из них. Какое число запомнили?

- 1) 45
- 2) 60
- 3) 70
- 4) 90

19 1 2 3 4

19. Исполнитель РОБОТ может передвигаться на одну клетку **вверх**, **вниз**, **вправо** и **влево** прямоугольного клетчатого поля, на котором расположены горизонтальные и вертикальные стенки. Двигаться вперед он может только тогда, когда стенка перед ним нет.

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

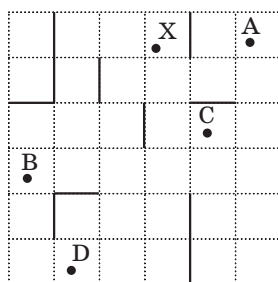
сверху свободно	снизу свободно	слева свободно	справа свободно
-----------------	----------------	----------------	-----------------

```

Цикл
ПОКА < условие > делать
  команда1
...
  командаN
Конец

```

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



Выберите точку, из которой РОБОТ начал движение, если он остановился в клетке X, перемещаясь по следующему алгоритму:

Пока (справа свободно) **делать**

Если снизу свободно **то**
идти вниз

Иначе

идти вправо

Если сверху свободно **то**
идти вверх

конец

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

20. Исполнитель РОБОТ может передвигаться на одну клетку **вверх**, **вниз**, **вправо** и **влево** прямоугольного клетчатого поля, на котором расположены горизонтальные и вертикальные стенки. Двигаться вперед он может только тогда, когда стенок перед ним нет.

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

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

Цикл

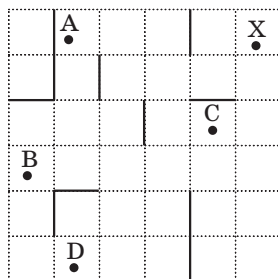
ПОКА *< условие >* **делать**
команда1

...

командаN

Конец

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



Выберите точку, из которой РОБОТ начал движение, если он остановился в клетке X, перемещаясь по следующему алгоритму:

Пока (сверху свободно) **делать**

Если справа свободно **то**

идти вправо

Иначе

идти вверх

Если сверху свободно **то**

идти вверх

конец

1) A

2) B

3) C

4) D

21 1 2 3 4

21. Исполнитель РОБОТ может передвигаться на одну клетку **вверх**, **вниз**, **вправо** и **влево** прямоугольного клетчатого поля, на котором расположены горизонтальные и вертикальные стенки. Двигаться вперед он может только тогда, когда стенок перед ним нет.

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

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

Цикл

ПОКА < условие > **делать**

команда1

...

командаN

Конец

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

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

НАЧАЛО

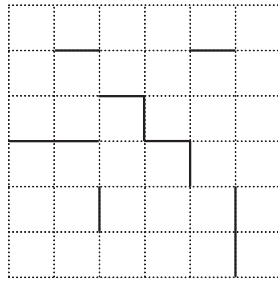
ПОКА < сверху свободно > **вверх**

ПОКА < справа свободно > **вправо**

ПОКА < снизу свободно > **вниз**

ПОКА < слева свободно > **влево**

КОНЕЦ



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

22. Исполнитель РОБОТ может передвигаться на одну клетку **вверх, вниз, вправо и влево** прямоугольного клетчатого поля, на котором расположены горизонтальные и вертикальные стенки. Двигаться вперед он может только тогда, когда стенок перед ним нет, иначе РОБОТ разрушается.

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

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

```

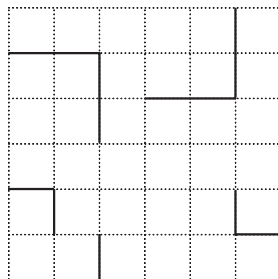
Цикл
ПОКА < условие > делать
команда1
...
командаN
Конец
  
```

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

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

```

НАЧАЛО
ПОКА < снизу свободно > вниз
ПОКА < слева свободно > влево
ПОКА < сверху свободно > вверх
ПОКА < справа свободно > вправо
КОНЕЦ
  
```



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

23. Исполнитель РОБОТ может передвигаться на одну клетку **вверх**, **вниз**, **вправо** и **влево** прямоугольного клетчатого поля, на котором расположены горизонтальные и вертикальные стенки. Двигаться вперед он может только тогда, когда стенок перед ним нет.

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

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

```

Цикл
ПОКА < условие > делать
  команда1
...
  командаN
Конец
  
```

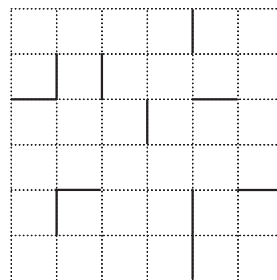
выполняется, пока условие истинно, иначе происходит переход на следующую строку.

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

Примечание: Если РОБОТ выполнит движение в сторону стены, то он разрушится, а выполнение программы прервется.

```

НАЧАЛО
ПОКА < слева свободно > влево
ПОКА < снизу свободно > вниз
ПОКА < справа свободно > вправо
ПОКА < сверху свободно > вверх
КОНЕЦ
  
```



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

24. Исполнитель РОБОТ может передвигаться на одну клетку **вверх**, **вниз**, **вправо** и **влево** прямоугольного клетчатого поля, на котором расположены горизонтальные и вертикальные стенки. Двигаться вперед он может только тогда, когда стенок перед ним нет. Команда Крась закрашивает клетку, в которой стоит РОБОТ.

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

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

```

Цикл
ПОКА < условие > делать
  команда1
...
  командаN
Конец

```

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

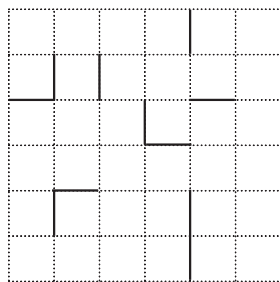
Сколько клеток (K_1) приведенного лабиринта соответствуют требованию, что, выполнив предложенную ниже программу, РОБОТ остановится в той же клетке, с которой он начал движение? Сколько клеток (K_2) приведенного лабиринта будут при этом закрашены? Сколько в сумме таких клеток (K_1+K_2)?

Примечание: Если РОБОТ выполнит движение в сторону стены, то он разрушится, а выполнение программы прервется.

```

НАЧАЛО
ПОКА < слева свободно > влево крась
ПОКА < снизу свободно > вниз крась
ПОКА < справа свободно > вправо крась
ПОКА < сверху свободно > вверх крась
КОНЕЦ

```



- 1) 19
- 2) 16
- 3) 10
- 4) 4

► Во второй части экзаменационной работы располагаются задания, ответ на которые необходимо получить самостоятельно и записать в специально отведенное для этого место.

При выполнении этих заданий также можно тренироваться в заполнении бланков, предлагаемых на экзамене (№ 1 и № 2). В специально отведенном месте, рядом с каждым заданием, расположен фрагмент бланка, который можно заполнять. По окончании работы следует перенести все ответы в образцы настоящих бланков.

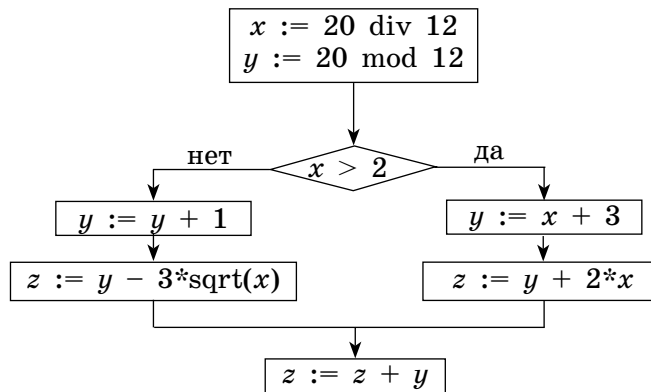
25

25. Запишите значение переменной **f** после выполнения фрагмента алгоритма для переменной целого типа $x = 1056$

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
$B = x \bmod 100 \setminus 10$ $C = x \bmod 10$ $D = x \setminus 1000$ $Y = 100 * c + 10 * d + b$ $f = x - y$	$B := x \bmod 100 \operatorname{div} 10;$ $C := x \bmod 10;$ $D := x \operatorname{div} 1000;$ $Y := 100 * c + 10 * d + b;$ $f := x - y;$	$B :=$ $\operatorname{div}(\operatorname{mod}(x, 100), 10)$ $C := \operatorname{mod}(x, 10)$ $D := \operatorname{div}(x, 1000)$ $Y := 100 * c + 10 * d + b$ $f := x - y$

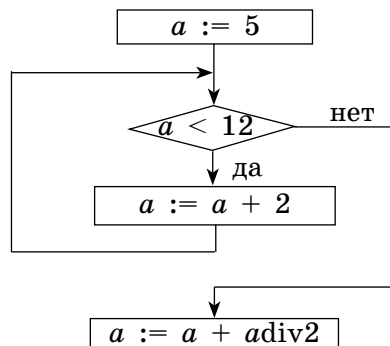
26

26. Запишите значение переменной **z** после выполнения фрагмента алгоритма



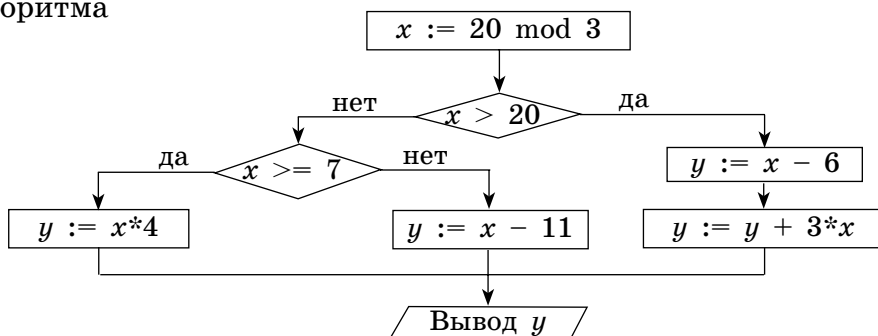
27

27. Запишите значение переменной **z** после выполнения фрагмента алгоритма



28. Запишите значение переменной y после выполнения фрагмента алгоритма

28



где $x \bmod y$ — операция mod — получение остатка от деления нацело x на y ;

29. Запишите значение переменной f после выполнения фрагмента алгоритма для переменной целого типа $x = 1221$.

29

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
$A = x \operatorname{div} 1000$ $B = x \operatorname{mod} 1000 \setminus 100$ $C = x \operatorname{mod} 100 \setminus 10$ $D = x \operatorname{mod} 10$ $Y = 1000*d + 100*c + 10*b + a$ $f = x - y + 1$	$A := x \operatorname{div} 1000;$ $B := x \operatorname{mod} 1000 \operatorname{div} 100;$ $C := x \operatorname{mod} 100 \operatorname{div} 10;$ $D := x \operatorname{mod} 10;$ $Y := 1000*d + 100*c + 10*b + a;$ $f := x - y + 1;$	$A := (x, 1000)$ $B := \operatorname{div}(\operatorname{mod}(x, 1000), 100)$ $C := \operatorname{div}(\operatorname{mod}(x, 100), 10)$ $D := \operatorname{mod}(x, 10)$ $Y := 1000*d + 100*c + 10*b + a$ $f := x - y + 1$

30. Запишите значение переменной f после выполнения фрагмента алгоритма для переменной целого типа $x = 1234$.

30

БЕЙСИК	ПАСКАЛЬ	ПСЕВДОКОД
$A = x \operatorname{d} \setminus 1000$ $B = x \operatorname{mod} 1000 \setminus 100$ $C = x \operatorname{mod} 100 \setminus 10$ $D = x \operatorname{mod} 10$ $Y = 1000*d + 100*c + 10*b + a$ $f = x - y + 1$	$A := x \operatorname{div} 1000;$ $B := x \operatorname{mod} 1000 \operatorname{div} 100;$ $C := x \operatorname{mod} 100 \operatorname{div} 10;$ $D := x \operatorname{mod} 10;$ $Y := 1000*d + 100*c + 10*b + a;$ $f := x - y + 1;$	$A := (x, 1000)$ $B := \operatorname{div}(\operatorname{mod}(x, 1000), 100)$ $C := \operatorname{div}(\operatorname{mod}(x, 100), 10)$ $D := \operatorname{mod}(x, 10)$ $Y := 1000*d + 100*c + 10*b + a$ $f := x - y + 1$

31. Исполнитель Кузнечик живет на числовой оси и имеет следующую систему команд:

31

Вперед a — Кузнечик прыгает вправо на a единиц;

Назад b — Кузнечик прыгает назад на b единиц.

Переменные a и b могут принимать любые целые положительные значения. Кузнечик из некоторой точки, выполняя только команды

1) Вперед 3

2) Назад 2,

оказался в точке 10. Из какой точки он начал движение, если выполнил 5 команд, а первой его командой была команда Вперед 3? Если таких точек несколько, то перечислите их через запятую в порядке возрастания.

32. С помощью двух сосудов А объемом 3 литра и В объемом 8 литров требуется отмерить 5 литров, причем вначале оба сосуда пусты и допустимы только следующие действия:

если сосуд А пустой, то наполнить его до краев;
 сосуд А перелить в сосуд В;
 если сосуд В наполнен до краев, то освободить его так, чтобы он остался пустым.

Сколько раз будет наполняться сосуд В до краев прежде, чем будет отмерен требуемый объем в литрах (начать с заполнения сосуда В)?

33. Исполнитель Вычислитель имеет следующую систему команд:

- 1) к числу прибавить 4
- 2) число умножить на 3.

Запишите порядок команд в программе получения из числа 11 числа 167, содержащей не более 7 команд, указывая лишь номера команд.

(Например, программа **21211** — это программа для исполнителя с командами 1) к числу прибавить 3 и 2) число умножить на 4.

умножь на 4
 прибавь 3
 умножь на 4
 прибавь 3
 прибавь 3,
 которая преобразует число 2 в число 50.)

34. Исполнитель Черепашка перемещается на экране компьютера, оставляя след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют две команды:

Вперед n , где n — целое число, вызывает передвижение Черепашки на n шагов в направлении движения.

Направо m , где m — целое число, вызывает изменение направления движения на m градусов по часовой стрелке (от 0° до 359°).

Запись **Повтори 5 [Команда1 Команда2]** означает, что последовательность команд в скобках повторится 5 раз.

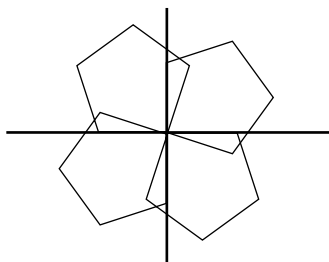
Исполнитель интерпретирует эту запись как одну команду.

Черепашке был дан для исполнения следующий алгоритм:

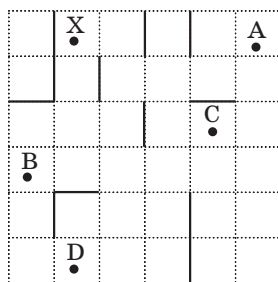
Повтори 4 [Повтори 5 [Вперед 100 Направо X] Направо Y]

На экране появилась следующая фигура (одна из сторон каждого пятиугольника совпадает с одной из двух перпендикулярных осей, а одна из вершин каждого пятиугольника находится в точке пересечения осей).

Определить и записать в ответе значения X и Y через запятую.



35. Исполнитель РОБОТ может передвигаться на одну клетку **вверх**, **вниз**, **вправо** и **влево** прямоугольного клетчатого поля, на котором расположены горизонтальные и вертикальные стенки. Двигаться вперед он может только тогда, когда стенок перед ним нет, иначе он разрушается.



Сколько клеток пройдет РОБОТ, перемещаясь по следующему алгоритму из точки D (считать исходную и конечную точки движения):

Пока (слева свободно) **делать**
Если сверху свободно **то**
 идти вверх
Иначе
 идти влево
Если слева свободно **то**
 идти влево
Иначе
 идти вправо
Если сверху свободно **то**
 идти вверх
 конец

36. Исполнитель Черепашка перемещается на экране компьютера, оставляя след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют две команды:

Вперед n , где n — целое число, вызывает передвижение Черепашки на n шагов в направлении движения.

Направо m , где m — целое число, вызывает изменение направления движения на m градусов по часовой стрелке (от 0° до 359°).

Запись **Повтори 5 [Команда1 Команда2]** означает, что последовательность команд в скобках повторится 5 раз.

Исполнитель интерпретирует эту запись как одну команду.

35

36

Черепашке был дан для исполнения следующий алгоритм рисования окружности:

Повтори 360[Вперед 1 Направо 1]

Чтобы нарисовать полуокружность в приведенном ниже алгоритме, укажите через запятую значения x, y, z .

Повтори x [Вперед y Направо z]

37. Исполнитель Кузнечик живет на числовой оси и имеет следующую систему команд:

Вперед a — Кузнечик прыгает вправо на a единиц;

Назад b — Кузнечик прыгает назад на b единиц.

Переменные a и b могут принимать любые целые положительные значения. Кузнечик из некоторой точки, выполняя только команды

1. Вперед 3

2. Назад 2,

оказался в точке 10. Из скольких точек он начал движение, если выполнил 5 команд, а первой его командой была команда Вперед 3? В ответе укажите через запятую количество точек и последовательность команд для самой большей по значению точки. Если таких последовательностей несколько, приведите их все. Например: 3,11221,12121, т.е. 3 точки и две последовательности команд 11221 и 12121.

► В третьей части экзаменационной работы располагаются самые сложные и трудоемкие задания, на которые необходимо представить развернутое решение или обоснование полученного ответа. Полное решение и полученный ответ записываются четко и разборчиво в бланк ответов № 2.

Для эффективной подготовки к экзамену рекомендуем предлагаемые задачи выполнять на компьютере на том языке программирования, который изучался. Надо обязательно исследовать работу программы с различными исходными данными (максимальное, минимальное значение, равенство нулю, положительные и отрицательные значения переменных и т.п.), а также запускать выполнение программы в режиме отладки и анализировать значения переменных, участвующих в проверке тех или иных условий. Будет нелишним там, где это возможно, использовать подпрограммы-функции и подпрограммы-процедуры. Необходимо понимать разницу между глобальными и локальными переменными, входными и выходными данными процедур.

Старайтесь использовать комментарии в программах, чтобы после написания программы выполнить ее проверку, облегчить чтение программы проверяющим экспертам.

Многие задания можно выполнять несколькими способами. Выберите наиболее эффективный из них, требующий меньше времени, например на обработку массивов, или менее трудоемкий в смысле программирования. Задания раздела С4 можно решать как с ис-

пользованием записей (record), так и с помощью массивов. Обращайте внимание на типы переменных, анализируйте, какие типы целесообразно использовать (например, longint, integer, byte, word) для экономии памяти. Разными способами можно вводить данные в задачах, если это не оговорено особо (с клавиатуры, с помощью генератора случайных чисел, из файла). Также разными способами можно выполнять вывод результатов. Если не оговорено особо, выбирайте более простые способы, чтобы не допустить ошибок.

Пишите тексты программ четко и разборчиво, чтобы их легко было понять и проверить.

Задания С1

1. Требовалось написать программу вычисления НОД (наибольший общий делитель) трех целых чисел. Программист написал программу с ошибкой.

1

ПАСКАЛЬ	БЕЙСИК (другая ошибка)
<pre> Var a,b,c,nod1,m:integer; Procedure nod(x,y,z:integer); Var n:integer; Begin z:=x;n:=y; While z<>n do begin If z>n then z:=z-n else n:=n-z; end; end; Begin Writeln('Введите три целых числа'); Readln(a); Readln(b); Readln(c); Readln(m); Nod(a,b,nod1); Nod(c,nod1,m); Writeln ('НОД=NOD=', m); readln; End.</pre>	<pre> Defint a,b,c,m,nod1 cls Input «a=»;a Input «b=»;b Input «c=»;c Input «m=»;m Call nod(a,b,nod1) Call nod(c, m,nod1) Print «НОД=»;m End Sub nod(x as integer,y as integer,z as integer) z=x:n=y While z <> n If z>n then z=z-n else n=n-z End if Wend End sub</pre>

- Привести пример таких исходных данных a, b и c, для которых программа дает неправильный результат.
 - Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.
 - Укажите, какая часть программы является лишней.
2. Требовалось написать программу подсчета мест багажа, загружаемого в багажный отсек самолета, чтобы его вес не превысил допустимый.

2

Программист написал программу с ошибкой.

ПАСКАЛЬ	БЕЙСИК
<pre> Var m:integer; B,s, p_max:real; Begin Write('Введите допустимый сум- марный вес='); Readln(p_max); Readln(m); M:=0;s:=0; While s>=p_max do begin Write('Введите вес='); Readln(b); M:=m+1; S:=s+b; end; Writeln ('Количество мест бага- жа=', m); Readln; End. </pre>	<pre> Defint m Input «Введите допустимый сум- марный вес=»; pmax Input «m=»;m M=0;s=0 While s>=pmax Input «Введите вес=»;b M=m+1 S=s+b Wend Print «Количество мест багажа=»;m End </pre>

- Привести пример таких исходных данных **p_max** и **b**, для которых программа дает неправильный результат.
- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.
- Укажите, какая часть программы является лишней.

3. Требовалось написать программу определения, является ли число простым (простым называется число, которое делится только на единицу и само себя). Программист написал программу с ошибкой.

ПАСКАЛЬ	БЕЙСИК
<pre> Var A, K,i:integer; Begin Writeln('Введите целое число'); Readln(A); K:=0; For i:=2 to a-1 do If a mod i<>0 then k:=k+1 If k=0 then Writeln(a, ' - число простое') else Writeln(a, ' - число составное') Readln; end. </pre>	<pre> Defint A,k,i Input «Введите целое число»;A K=0 For i=2 to a-1 If a mod i<>0 then k=k+1 End if next i If k=0 then Print a;» - число простое « else Print a;» - число составное» End if end </pre>

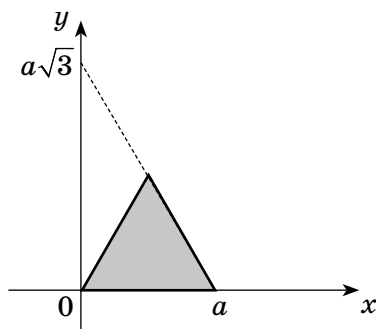
- Привести пример такого **A**, для которого программа дает неправильный результат.
- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.
- Укажите, какая часть программы является лишней.

4. Требовалось написать программу определения по номеру года, является ли он високосным. Год является високосным, если его номер делится на 4 без остатка, но для номера, который оканчивается на два нуля, он должен делиться без остатка на 400. Программист написал программу с ошибкой.

ПАСКАЛЬ	БЕЙСИК
<pre>Const Res1=' у – високосный год'; Const Res2=' п – невисокосный год'; Var A:integer; Begin Writeln('Введите год'); Readln(A); If (a mod 100=0) and (a mod 400=0) then writeln(a,res1) else writeln(a,res2); Begin If a mod 4=0 then writeln(a,res1) else writeln(a,res2); End; Readln; end.</pre>	<pre>Defint A Defstr res1,res2 Input «Введите год»;A Res1=« – високосный год» Res2=« – невисокосный год» If (a mod 100=0) and (a mod 400= 0) then print a;res1 else print a;res1 End if If a mod 4=0 then print a;res1 else print a;res1 End if end</pre>

- Привести пример такого **A**, для которого программа дает неправильный результат.
- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

5. Требовалось написать программу определения количества точек из N точек, сгенерированных случайным образом, попавших в равносторонний треугольник со стороной a , включая его границы (см. рисунок к задаче). Программист написал программу с ошибкой.



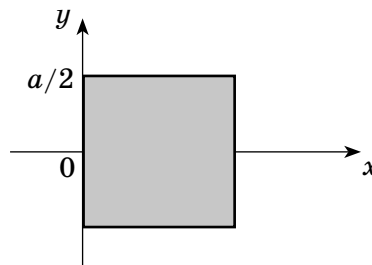
ПАСКАЛЬ	БЕЙСИК
<pre>Const n=100; Var I,k:integer; A,x,y:real; Begin Randomize; Writeln('Введите сторону треу- гольника');</pre>	<pre>Const n=100 Defint I,k Randomize Input «Введите сторону треуголь- ника»;A Input «x, y»;x,y K=0;</pre>

ПАСКАЛЬ	БЕЙСИК
<pre> Readln(A); Readln(x,y); K:=0; For i:=1 to n do begin X:=-a/2+ random*a*2; Y:=-a/2+ random*a*2; writeln('x=',x:5:2, ' y=',y:5:2); If (x>=0) and (x<=a*sqrt(3)) and (y>=0) and (y<=sqrt(3)*(a-x)) then k:=k+1; End; Writeln(k,' точек попало внутрь треугольника '); Readln; end.</pre>	<pre> For i=1 to n do X=-a/2+ rnd*2*a Y= -a/2+ rnd*2*a Print «x=»;x;»y=»;y If (x>=0) and (x<=a*sqr(3)) and (y>=0) and (y<=sqr(3)*(a-x)) then k=k+1 End if End i Print k ;» точек попало внутрь треугольника» end</pre>

- Привести пример таких x, y , для которых программа дает неправильный результат.
- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.
- Укажите, какая часть программы является лишней.

6

6. Требовалось написать программу определения количества точек из N точек, сгенерированных случайным образом, попавших в квадрат со стороной a , включая его границы (см. рисунок к задаче). Программист написал программу с ошибкой.

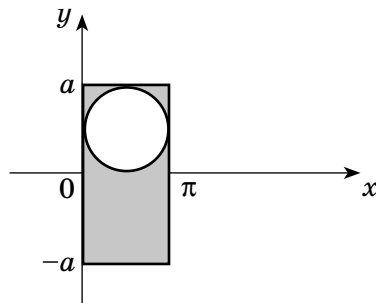


ПАСКАЛЬ	БЕЙСИК
<pre> Const n=100; Var I,k:integer; A,x,y:real; Begin Randomize; Writeln('Введите сторону квадра- та'); Readln(A); Writeln('Введите координаты точ- ки x,y'); Readln(x,y); K:=0; For i:=1 to n do begin X:=-a/2+ random*2*a; Y:= -a/2+ random*2*a;</pre>	<pre> Const n=100 Defint I,k Randomize Input «Введите сторону квadrата»;A Input «x, y»;x,y K=0; For i=1 to n do X=-a/2+ RND*2*a Y= -a/2+ RND*2*a Print «x=»;x;»y=»;y If (x>=0) and (x<=a) and (y<=a/2) then k=k+1 End if next i</pre>

ПАСКАЛЬ	БЕЙСИК
<pre> Writeln('x=',x:5:2,' y=',y:5:2); If (x>=0) and (x<=a) and (y<=a/2) then k:=k+1; End; Writeln(k,' точек/точки попало внутри квадрата'); Readln; end. </pre>	<pre> Print k ;» точек/точки попало внутри квадрата» end </pre>

- Привести пример точки с координатами (x, y) , для которой программа дает неправильный результат.
- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.
- Укажите, какая часть программы является лишней.

7. Требовалось написать программу, чтобы определить, попадет ли точка с координатами (x, y) в заштрихованную область, ограниченную окружностью с центром $(\pi/2, a/2)$, с диаметром a ($a < \pi$) и сторонами прямоугольника $(2a, \pi)$, учитывая границы. Программист написал программу с ошибкой.



Паскаль	Бейсик
<pre> Var a,x,y,z:real; Begin Writeln('введите a, x,y'); Readln(a,x,y); z:=(x-pi/2)*(x-pi/2)+(y-a/2)*(y-a/2); If (x>=0) and (x<=pi) and (y<=a) and (abs(z)>=a*a/4) then writeln('Точка принадлежит области') else writeln('Точка не принадлежит области'); Readln; End. </pre>	<pre> INPUT «Введите a,x, y: «;a,x,y Pi=3.14159265358 z=(x-pi/2)*(x-pi/2)+(y-a/2)*(y-a/2) If (x>=0) and (x<=pi) and (y<=a) and (abs(z)>=a*a/4) then print «Точка принадлежит области» else print «Точка не принадле- жит» End if End </pre>

- Привести пример таких a, x, y , для которых программа дает неправильный результат.
- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.
- Укажите, какая часть программы является лишней.

8. Требовалось написать программу, чтобы прочитать из файла *input.dat* строку и проверить, является ли она записью восьмеричного числа. Вывести сообщение, является ли число восьмеричным. Программист написал программу с ошибкой.

ПАСКАЛЬ	БЕЙСИК
<pre> program file_my; uses crt; Var txt:text; Str:string; {для введенной строки} Ch:char; I,k,n,x,j:integer; Begin Assign(txt,'input.dat'); {Связываем переменную txt с внешним файлом и открываем файл для чтения} Reset(txt); Readln(txt,str); {считываем из файла} Close(txt); {закрываем файл} N:=length(str); For i:=1 to n do begin j:=0; X:=ord(str[i])-ord('0') ; For k:=1 to 7 do begin If x = k then j:=j+1; end; {конец цикла по k} if j=0 then begin writeln('Число записано не в восьмеричной системе счисления'); readln; exit; end; {конец if} end; {конец цикла по i} writeln('Число записано в восьмеричной системе счисления'); Readln; End. </pre>	<pre> DIM s\$(255) as string Defint I,k,n,x,j Defstr A,B,m,c 'или m\$ A:=«input.dat» B:=«output.dat» 'Открываем файл A для чтения как 1 OPEN a FOR INPUT AS #1 INPUT\$ #1,m 'm-набор символов. Читает строку символов из файла CLOSE 'закрываем файл N=len(m) For i=1 to n S\$(i)= mid\$(m,I,1) Next i For i=1 to n J=0 For k=1 to 7 X=val(s\$(i)) If x = k then j=j+1 Next k If j=0 then Print «Число записано не в восьмеричной системе счисления» exit End if Next i Print «Число записано в восьмеричной системе счисления» end </pre>

- Привести пример такой строки, для которой программа дает неправильный результат.
- Исправьте программу так, чтобы в ней не выполнялось лишних операций.
- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

9. Требовалось написать программу вычисления числа сочетаний из n элементов по m , используя функцию вычисления факториала. Программист написал программу с ошибкой.

$$C_n^m = \frac{n!}{m!(n-m)!}$$

ПАСКАЛЬ	БЕЙСИК
<pre> Var c,f1,f2,f3:longint; n,m:integer; {функция вычисления факториала F числа n} {F=1*2*3*...*n, для n=0 и n=1 F=1} Function fact(n:integer):longint; Var K:integer; F:longint; Begin If n=1 or n=0 then f:=1 Else begin F:=1; For k:=2 to n do F:=f*k; Fact:=f; end; End; {основная программа} begin wirte('Введите n='); readln(n); wirte('Введите m<n='); readln(m); f1:=fact(n); f2:=fact(m); f3:=fact(n-m); c:=f1 div (f2*f3); writeln('Число сочетаний из n по m C=',c:6:0); readln; End. </pre>	<pre> Deflng c,f1,f2,f3 Defint n,m Rem функция вычисления факто- риала F числа n Rem F=1*2*3*...*n, для n=0 и n=1 F=1 DEF FNfact&(n) If n=1 or n=0 then f&=1 Else F&=1 For k=2 to n F&=f&*k FNFact&=f& End if End i END DEF Input «Введите n=»;n Input «Введите m<n=»;m f1=FNfact&(n) f2=FNfact&(m) f3=FNfact&(n-m) c=f1 \ (f2*f3) print «Число сочетаний из n по m C=';c end </pre>

- Привести пример таких значений n и m , для которых программа дает неправильный результат.
- Исправьте программу так, чтобы в ней не выполнялось лишних операций.
- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

Задания С2

1. Опишите на русском языке или одном из языков программирования алгоритм определения индексов элементов двумерного массива A размером $N \times N$, равных максимальному элементу, и их количества.
2. Опишите на русском языке или одном из языков программирования алгоритм перестановки элементов в каждом из столбцов целого двумерного массива A размером $N \times M$ в обратном порядке.
3. Опишите на русском языке или одном из языков программирования алгоритм определения количества элементов столбца двумерного массива A размером $N \times N$, равных максимальному, в каждом столбце. Поместите их в отдельный массив.

 1

 2

 3

4 5 6 1

4. Массив содержит несколько предложений, разделенных точкой, вопросительным или восклицательным знаками. Опишите на русском языке или одном из языков программирования алгоритм вывода на экран количества предложений.
5. Опишите на русском языке или одном из языков программирования алгоритм определения суммы элементов в строках массива A размером $N \times M$ и его столбцах. Сохраните эти суммы в одном массиве: сначала суммы строк, затем суммы столбцов.
6. Опишите на русском языке или одном из языков программирования алгоритм поиска элементов целого массива X размером N , являющихся простыми числами. Определите их количество и сохраните в отдельном массиве. Элементы массива имеют значения от 0 до 50.

Задания С4

1. На вход программе подаются сведения о филиалах некоторой фирмы. В первой строке сообщается количество филиалов N , которое больше 5, но меньше или равно 50, каждая из следующих N строк имеет следующий формат: <Название филиала> и количество произведенной продукции в данном квартале по месяцам: <Месяц1> < Месяц 2> < Месяц 3>, где <Название филиала > — строка, состоящая не более чем из 10 символов, <Месяц1> < Месяц 2> < Месяц 3> — целые числа от 0 до 5000000. <Название филиала > <Месяц1> < Месяц 2> < Месяц 3> разделены одним пробелом.

Пример входных строк:

```
25
Филиал1 500000 400400 500000
Филиал2 50000 800400 70000
Филиал3 200000 60400 90000
Филиал4 505000 34400 670000
```

Требуется написать программу, которая определит по два филиала, которые выпустили наибольшее и наименьшее количество продукции за квартал (указать произведенное количество). Если среди филиалов есть филиалы, выпустившие такое же количество продукции, что 2 лучших или 2 худших, то вывести и их названия. Определить разницу между максимальным и минимальным выпуском за квартал.

Пример выходных строк:

```
Лучшие филиалы:
Филиал2      3900000
Филиал12     3600050
```

Филиалы-аутсайдеры:

Филиал3 1200000
Филиал14 1100500
Филиал 17 1009000

Разница Max-Min=1891000

2. На вход программе подаются сведения об учениках школы (не более 300 чел.). В первой строке сообщается количество учащихся N , каждая из следующих N строк имеет формат: <Фамилия> <Имя> <Отчество> <Пол>, <Год рождения>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 12 символов, <Отчество> строка, состоящая не более чем из 12 символов, <Пол> — 1 символ (например, м или ж), <Год рождения> — целое четырехзначное число. <Фамилия> <Имя> <Отчество> <Пол>, <Год рождения> разделены одним пробелом.

Пример входных строк:

150

Кузнецов Сергей Иванович м 1995
Светлов Андрей Петрович м 1997
Андреева Анна Сергеевна ж 1996
Кузнецов Сергей Дмитриевич м 1995
Петрова Елена Антоновна ж 1998

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая выведет отдельно списки девочек и мальчиков в порядке возрастания их года рождения.

Пример выходных строк:

Девочки

Андреева Анна Сергеевна 1996
Петрова Елена Антоновна 1998

Мальчики

Кузнецов Сергей Иванович 1995
Кузнецов Сергей Дмитриевич 1995
Светлов Андрей Петрович 1997

3. На вход программе подаются сведения об абитуриентах некоторого вуза. В первой строке сообщается количество абитуриентов N , которое больше 20, но меньше или равно 100, каждая из следующих N строк имеет следующий формат: <Фамилия> <Имя> <Отчество> <Балл1> <Балл 2> <Балл 3>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 15 символов, <Отчество> — строка, состоящая не более чем из 20 символов, <Балл1> <Балл 2> <Балл 3> — целое число от 0 до 100. <Фамилия>, <Имя> и

<Отчество>, а также <Отчество> и <Балл1> < Балл 2> < Балл 3> разделены одним пробелом. Входные данные не упорядочены никаким способом.

Пример входных строк:

```
45
Федорова Ирина Евгеньевна 80 56 76
Иванов Сергей Захарович 59 77 90
Светлов Андрей Иванович 89 67 80
```

Требуется написать программу, которая определит сумму баллов абитуриентов и будет выводить на экран их фамилии, имена, отчества и сумму баллов в порядке убывания. Список пронумеровать.

Пример выходных строк:

```
1. Светлов Андрей Иванович 236
2. Иванов Сергей Захарович 226
3. Федорова Ирина Евгеньевна 212
```

4

4. На вход программе подаются сведения о пациентах больницы одного года. В первой строке сообщается количество пациентов N, которых больше 10, но не более 300, каждая из следующих N строк имеет следующий формат: <Фамилия> <Имя> <Отчество> <Номер истории болезни> <Пол> <Год рождения> <Дата поступления> <Дата выписки>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 15 символов, <Отчество> — строка, состоящая не более чем из 15 символов, <Номер истории болезни> — целое число, <Пол> — буква “м” или “ж”, <Год рождения> — целое число, <Дата поступления> <Дата выписки> имеют формат чч.мм, где чч — число, мм — номер месяца разделены точкой (например, 04.12). <Фамилия>, <Имя>, <Отчество>, <Номер истории болезни>, <Пол>, <Год рождения>, <Дата начала болезни>, <Дата окончания болезни> разделены одним пробелом.

Пример входной строки:

```
200
Федорова Ирина Павловна 345 ж 1967 04.10 20.10
Иванов Сергей Петрович 1029 м 1940 12.12 23.12
Петров Сергей Иванович 4567 м 1993 30.04 10.05
Волохова Зинаида Евгеньевна 345 ж 1992 30.06 10.08
```

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая определит, сколько дней находился в больнице каждый из пациентов.

Вывести номер истории болезни, фамилию, инициалы пациентов и количество дней. Список вывести в алфавитном порядке. Исходные данные вводятся корректно.

Результат работы программы может выглядеть так:

Номер ИБ	Фамилия И.О.	Количество дней
1234	Волохова З.Е.	11
2345	Иванов С.П.	12
123	Петров С.И.	11
222	Федорова И.П.	42

5. На вход программе подаются сведения об учениках школы (не более 300 чел.) В первой строке сообщается количество учащихся N, каждая из следующих N строк имеет формат: <Фамилия> <Имя> <название класса><Физкультурная группа>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 12 символов, <номер класса> — целое число (например, 11, 9, 10), <Физкультурная группа> — целое число (1, 2 или 3), <Фамилия>, <Имя>, <Номер класса>, <Физкультурная группа> — разделены одним пробелом.

Пример входных строк:

150

Кузнецов Сергей 9 3

Светлов Андрей 11 2

Платонов Сергей 9 1

Сидоров Юрий 11 1

Петрова Елена 10 2

Иванов Дмитрий 9 3

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая определит количество учащихся в каждой группе и выведет список учащихся по физкультурным группам начиная с 1-й. Списки пронумеровать, алфавитный порядок в группах можно не соблюдать.

Пример выходных строк:

1 физкультурная группа — 2 уч-ся.

1. Платонов Сергей 9 кл.

2. Сидоров Юрий 11 кл.

2 физкультурная группа — 2 уч-ся.

Светлов Андрей 11 кл.

Петрова Елена 10 кл.

3 физкультурная группа — 2 уч-ся.

1. Кузнецов Сергей 9 кл.

2. Иванов Дмитрий 9 кл.

6. На вход программе подаются сведения об учениках школы (не более 300 чел.) В первой строке сообщается количество учащихся N, каждая из следующих N строк имеет формат: <Фамилия>

<Имя> < класс> <Оценка1> <Оценка2 >, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 12 символов, <класс> — целое число (например, 11, 9, 10), <Оценка1> <Оценка2 > — целое число от 1 до 5. <Фамилия>, <Имя>, <класс>, <Оценка1>, <Оценка2 > разделены одним пробелом.

Пример входных строк:

```
150
Кузнецов Сергей 9 5 4
Светлов Андрей 10 5 3
Троекурова Маша 11 3 3
Андреев Сергей 9 4 3
Платов Иван 11 3 4
Денисов Олег 10 3 3
```

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая выведет список учащихся, упорядоченный по убыванию суммы баллов. Трех лучших отметить словом «лидеры». Если среди учащихся будут те, которые набрали тот же средний балл, что и лидеры, то отметить и их фамилии.

Пример выходных строк:

Фамилия имя	Класс	Сумма баллов
Кузнецов Сергей	9	9 лидер
Светлов Андрей	10	8 лидер
Андреев Сергей	9	7 лидер
Платов Иван	11	7 лидер
Троекурова Маша	11	6
Денисов Олег	10	6

ОТВЕТЫ

ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

№ задания	Ответ	№ задания	Ответ
1	1	13	4
2	4	14	3
3	3	15	2
4	4	16	3
5	2	17	3, 5, 7, 9
6	4	18	7
7	2	19	47
8	4	20	5
9	3	21	4
10	3	22	8, 8, 256
11	1	23	A
12	4		

ОСНОВЫ ЛОГИКИ

№ задания	Ответ	№ задания	Ответ
1	1	9	4
2	4	10	4
3	1	11	9
4	2	12	10
5	1	13	БОЗЛ
6	2	14	РФНБ
7	1	15	Г1, А4, В2, Ф3
8	4	16	ВКТШ

17. Р Е Ш Е Н И Е: Ответ не обязательно оформлять таблицей или иным способом. Можно просто описать словами, как будут ходить игроки. Эксперты проверят любое решение, и на конечном результате это не скажется.

Все камни из кучи удобно разложить (хотя бы мысленно) на 4 малые кучки, в первой оставив 2 камня, во второй и третьей — 4, а в четвертой — 1 камень. Первый игрок будет в выигрышной позиции, если первым ходом возьмет 2 камня, а после каждого хода противника будет брать столько камней, что в сумме они будут составлять 4 (т.е. за ход первого и второго игрока сумма взятых камней равна 4). Поэтому последний камень возьмет первый игрок и для него это и есть выигрышная стратегия.

18. Р Е Ш Е Н И Е: Выигрывает второй игрок.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны координаты фишки на каждом этапе игры.

Стартовая позиция	1-й ход	2-й ход	3-й ход	4-й ход
	I игрок (все варианты хода)	II игрок (выигрышный ход)	I игрок (все варианты хода)	II игрок (выигрышный ход, один из вариантов)
-3,2	0,5	<u>5,5</u>	8,8	<u>13,8</u>
			5,9	<u>10,9</u>
			10,5	<u>15,5</u>
	-3,6	<u>2,6</u>	5,9	<u>10,9</u>
			2,10	<u>7,10</u>
			7,6	<u>12,6</u>
	2,2	<u>5,5</u> или <u>2,6</u> (экзаменуемому достаточно привести один из вариантов)	Те же варианты третьего-четвертого ходов	

19. Р Е Ш Е Н И Е: Выигрывает первый игрок, своим первым ходом он должен поставить фишку в точке с координатами (3,4). Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке координаты фишки на каждом этапе игры.

1-й ход	2-й ход	3-й ход	4-й ход	5-й ход
Позиция после первого хода	II игрок (все варианты хода)	I игрок (выигрышный ход)	II игрок (все варианты хода)	I игрок (один из вариантов)
<u>3,4</u>	6,4	<u>6,6</u>	9,6	<u>12,6</u>
			6,8	<u>9,8</u>
			6,10	<u>9,10</u>
	3,6	<u>6,6</u>	Те же варианты 4—5 хода	
	3,8	<u>3,12</u>	Первый игрок выигрывает ответным ходом	

Таблица содержит *все возможные* варианты ходов второго игрока. Из нее видно, что при любом ответе второго игрока у первого имеется ход, приводящий к победе. [1]

20. Р Е Ш Е Н И Е: Выигрывает первый игрок, своим первым ходом он должен поставить фишку в точке с координатами (7, 12). Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке координаты фишки на каждом этапе игры.

1-й ход	2-й ход	3-й ход	Примечание
Позиция после первого хода	II игрок (все варианты хода)	I игрок (выигрышный ход)	
<u>7,12</u>	5,12	3,12 5,10 <u>5,9</u>	При безошибочной игре первый игрок выигрывает на 3-м ходу игры. (Подчеркнуты возможные выигрышные варианты.)
	7,10	5,10 <u>7,8</u> <u>7,7</u>	
	7,9	<u>5,9</u> <u>7,7</u> <u>7,6</u>	

ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ. МОДЕЛИРОВАНИЕ И КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

№ задания	Ответ	№ задания	Ответ
1	3	13	3
2	2	14	2
3	2	15	3
4	3	16	4
5	3	17	3
6	1	18	1
7	2	19	3
8	3	20	2
9	3	21	ВАЖГЕБД
10	2	22	ВГАБ
11	1	23	4123
12	4	24	38

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Ответы на задания части 1 с выбором ответа

№ задания	Ответ	№ задания	Ответ
1	3	7	2
2	1	8	4
3	2	9	2
4	1	10	3
5	4	11	3
6	3	12	3

Окончание табл.

№ задания	Ответ	№ задания	Ответ
13	4	19	4
14	2	20	2
15	1	21	2
16	4	22	3
17	4	23	3
18	3	24	1

Ответы на задания части 2 с записью ответа

№ задания	Ответ	№ задания	Ответ
25	441	32	1
26	15	33	1211211
27	19	34	72,90
28	-9	35	7
29	1	36	180, 1, 1
30	-3086	37	5, 12222
31	-5, 0, 5, 10, 15		

Пояснения-иллюстрации к задачам 19–24

<p>№ 19 Ответ номер 4.</p>	<p>№ 20 Ответ: точка В, т.е. ответ номер 2.</p>
<p>№ 21 Ответ номер 2.</p>	<p>№ 22 Ответ номер 3.</p>

<p align="center">№ 23 Ответ номер 3.</p>	<p align="center">№ 24 K1+K2=3+4+2+10=19 клеток, т.е. ответ номер 1.</p>

Ответы на задания С1

1.

Р Е Ш Е Н И Е

В программе используется процедура $\text{Nod}(x,y,z)$, где x и y — исходные два целых числа, z — результат, наибольший общий делитель. Для определения НОД трех чисел необходимо найти НОД первых двух, а потом НОД третьего числа и найденного НОД первых двух чисел. Например, найти НОД трех чисел: 10, 100, 15. $\text{НОД}(10,100)=10$, а $\text{НОД}(10,15)=5$. Искомый НОД трех чисел равен 5. В программе допущена ошибка при описании формальных параметров процедуры Nod .

ОТВЕТ на первый вопрос (Паскаль). При любых исходных данных программа будет давать ошибку, например: $a=12$, $b=16$, $c=20$, НОД не определяется. В процедуре происходит заикливание, т.к. неправильно объявлена переменная-результат z . Это параметр-переменная, перед ней должно стоять слово **Var**, которое указывает, что значение этого параметра передается в обоих направлениях: из программы в процедуру и наоборот. Параметры-значения x , y передаются только в одном направлении: из программы в процедуру. То есть после первого обращения к процедуре $\text{Nod}(a,b,\text{nod1})$ получим внутри процедуры правильный результат $\text{nod1}=4$, но он не будет передан в основную программу. Поэтому во втором обращении $\text{Nod}(c,\text{nod1},m)$ вместо 4 в качестве nod1 пойдет ноль, и процедура «заиклится».

ОТВЕТ на первый вопрос (Бейсик).

При любых исходных данных программа будет давать ошибку, например: $a=12$, $b=16$, $c=20$, $m=7$ НОД определяется неверно: равно 1.

Пояснение: После первого обращения к процедуре $\text{Nod}(a,b,\text{nod1})$ получим правильный результат $\text{nod1}=4$. Но второе обращение записано неверно **Call nod(c, m,nod1)** — неправильный порядок переменных. Пусть введено $m=7$, тогда *НОД для 20 и 7 будет 1*, тогда как правильный ответ для $a=12$, $b=16$, $c=20$ получим $m=4$.

Отсюда ответ на второй вопрос (Паскаль). Следует заменить список формальных параметров в описании процедуры **Procedure nod(x,y,z:integer);** на следующий

Procedure nod(x,y:integer; var z:integer). Ниже приведен текст программы с исправлением.

ОТВЕТ на второй вопрос (Бейсик).

Call nod(c, m,nod1) — неправильный порядок переменных, надо **Call nod(c, nod1, m)**. Так как из-за лишнего оператора **Input «m=»;m** переменная **m** может иметь любое значение, то ответ будет неправильным.

ОТВЕТ на третий вопрос (Паскаль, Бейсик). Ввод переменной **m** является лишним, т.к. это результат выполнения программы.

Правильная программа	
ПАСКАЛЬ	БЕЙСИК
<pre> Var a,b,c,nod1,m:integer; Procedure nod(x,y:integer; var z:integer); Var n:integer; Begin z:=x;n:=y; While z<>n do begin If z>n then z:=z-n else n:=n-z; end; end; Begin Writeln('Введите три целых чис- ла'); Readln(a); Readln(b); Readln(c); Nod(a,b,nod1); Nod(c,nod1,m); Writeln ('НОД=NOD=', m); readln; End. </pre>	<pre> DECLARE SUB nod (x AS INTEGER, y AS INTEGER, z AS INTEGER) DEFINT A-C, M-N CLS INPUT «a=»; a INPUT «b=»; b INPUT «c=»; c CALL nod(a, b, nod1) CALL nod(c, nod1, m) PRINT «nod=»; m END SUB nod (x AS INTEGER, y AS INTEGER, z AS INTEGER) z = x: n = y WHILE z <> n IF z > n THEN z = z - n ELSE n = n - z END IF WEND END SUB </pre>

2. Требовалось написать программу подсчета мест багажа, загружаемого в багажный отсек самолета, чтобы его вес не превысил допустимый.

Программист написал программу с ошибкой.

ПАСКАЛЬ	БЕЙСИК
<pre> Var m:integer; B,s, p_max:real; Begin Write('Введите допустимый сум- марный вес='); Readln(p_max); Readln(m); M:=0;s:=0; While s>=p_max do begin Write('Введите вес='); Readln(b); M:=m+1; S:=s+b; end; Writeln ('Количество мест бага- жа=', m-1); Readln; End. </pre>	<pre> Defint m Input «Введите допустимый сум- марный вес=»; pmax Input «m=»;m M=0;s=0 While s>=pmax Input «Введите вес=»;b M=m+1 S=s+b Wend Print «Количество мест багажа=»;m-1 End </pre>

- Привести пример таких исходных данных **p_max** и **b**, для которых программа дает неправильный результат.

О Т В Е Т: для любого положительного значения **p_max** цикл **While** выполняться не будет, т.к. условие **s >= p_max** будет ложным и не потребуется вводить значение переменной **b**. И хотя **p_max > 0**, ответ будет неверным: **Количество мест багажа=0**.

- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

О Т В Е Т: **While s >= p_max do** — ошибка, правильным будет **While s < p_max do**. Тогда для положительного значения **p_max** получим искомое количество мест багажа.

- Укажите, какая часть программы является лишней.

О Т В Е Т: **Readln(m);** — лишняя часть программы, **m** — искомое количество мест багажа, а не исходные данные.

Правильная программа	
ПАСКАЛЬ	БЕЙСИК
<pre> Var m:integer; B,s, p_max:real; Begin Write('Введите допустимый сум- марный вес='); Readln(p_max); M:=0;s:=0; While s<p_max do begin Write('Введите вес='); Readln(b); M:=m+1; S:=s+b; end; Writeln ('Количество мест бага- жа=', m-1); Readln; End.</pre>	<pre> DEFINT M INPUT «Введите допустимый сум- марный вес=»; pmax m = 0: s = 0 WHILE s < pmax INPUT «Введите вес=»; b m = m + 1 s = s + b WEND PRINT «Кол-во мест багажа=»; m - 1 END</pre>

3. Требовалось написать программу определения, является ли число простым (простым называется число, которое делится только на единицу и самое себя). Программист написал программу с ошибкой.

ПАСКАЛЬ	БЕЙСИК
<pre> Var A, K,i:integer; Begin Writeln('Введите целое число'); Readln(A); K:=0; For i:=2 to a-1 do If (a mod i)<>0 then k:=k+1; If k=0 then Writeln(a,' - число простое') else Writeln(a,' - число составное') Readln; end.</pre>	<pre> Defint A,k,i Input «Введите целое число»;A K=0 For i=2 to a-1 If (a mod i)<>0 then k=k+1 End if End i If k=0 then Print a;« - число простое » else Print a;« - число составное» End if end</pre>

- Привести пример такого **A**, для которого программа дает неправильный результат.

О Т В Е Т: **A** — любое составное число, например 14, т.к. $A \text{ Mod } 2$ будет равно 0, и $k=0$, вывод сообщения '14 — число простое'. Для любого простого числа, например 5, получим $k=3$ и вывод сообщения '5 — число составное'.

- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

О Т В Е Т: оператор **If (a mod i) <> 0 then k:=k+1;** заменить на оператор **If (a mod i)=0 then k:=k+1**, т.к. у простого числа остаток от деления на любое число не будет равен нулю, и переменная **k** останется равной нулю.

- Укажите, какая часть программы является лишней.

О Т В Е Т: Заголовок цикла **For i:=2 to a-1 do** предполагает лишние действия. Лучше написать **For i:=2 to a div 2 do**, чтобы сократить количество делений, т.к. не имеет смысла делить проверяемое число на числа, большие его половины.

Правильная программа	
ПАСКАЛЬ	БЕЙСИК
<pre> Var A, K,i:integer; Begin Writeln('Введите целое число'); Readln(A); K:=0; For i:=2 to a div 2 do If (a mod i)=0 then k:=k+1; If k=0 then Writeln(a,' - у число простое') else Writeln(a,' - по число составное'); Readln; end. </pre>	<pre> DEFINT A, I, K INPUT «Введите целое число»; a k = 0 FOR i = 2 TO a \ 2 IF (a MOD i) = 0 THEN k = k + 1 END IF NEXT IF k = 0 THEN PRINT a; « - число простое » ELSE PRINT a; « - число составное» END IF END </pre>

4. Требовалось написать программу определения по номеру года, является ли он високосным. Год является високосным, если его номер делится на 4 без остатка, но для номера, который оканчивается на два нуля, он должен делиться без остатка на 400. Программист написал программу с ошибкой.

ПАСКАЛЬ	БЕЙСИК
<pre> Const Res1=' у - високосный год'; Const Res2=' n - невисокосный год'; Var A:integer; Begin Writeln('Введите год'); Readln(A); </pre>	<pre> Defint A Defstr res1,res2 Input «Введите год»;A Res1=« - високосный год» Res2=« - невисокосный год» If (a mod 100=0) and I(a mod 400= 0) then print a;res1 else print a;res1 </pre>

ПАСКАЛЬ	БЕЙСИК
<pre>If (a mod 100=0) and (a mod 400=0) then writeln(a,res1) else writeln(a,res2); Begin If a mod 4=0 then writeln(a,res1) else writeln(a,res2); End; Readln; end.</pre>	<pre>End if If a mod 4=0 then print a;res1 else print a;res1 End if end</pre>

• Привести пример такого А, для которого программа дает неправильный результат.

О Т В Е Т: А=1000, т.к. это число делится на 100 без остатка и не делится на 400 без остатка, то получаем сообщение: **1000 — невисокосный год**, но, с другой стороны, 1000 делится без остатка на 4, и получаем сообщение: **1000 — високосный год**.

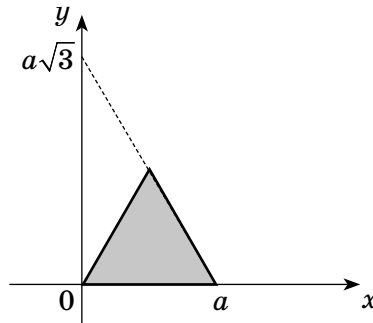
• Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

О Т В Е Т: для исправления ошибки следует использовать вложенный условный оператор **If**. Сначала надо проверить, оканчивается ли число на два нуля, а затем в случае истинности проверять его делимость без остатка на 400, а в противном случае на 4.

```
If (a mod 100=0) then {внешний if}
begin
  If (a mod 400=0) then writeln(a,res1) else writeln(a,res2);
End
Else {для внешнего If}
Begin
  If a mod 4=0 then writeln(a,res1) else writeln(a,res2);
End;
```

Правильная программа	
ПАСКАЛЬ	БЕЙСИК
<pre>Const Res1=' - високосный год'; Const Res2=' - невисокосный год'; Var A:integer; Begin Writeln('Введите год'); Readln(A); If (a mod 100=0) then begin if (a mod 400=0) then writeln(a,res1) else writeln(a,res2); end else Begin If a mod 4=0 then writeln(a,res1) else writeln(a,res2); End; Readln; end.</pre>	<pre>Defint A Defstr res1,res2 Input «Введите год»;A Res1=« - високосный год» Res2=« - невисокосный год» If (a mod 100=0) then If (a mod 400= 0) then print a;res1 else print a;res2 End if Else If a mod 4=0 then print a;res1 else print a;res2 End if End if end</pre>

5. Требовалось написать программу определения количества точек из N точек, сгенерированных случайным образом, попавших в равносторонний треугольник со стороной a , включая его границы (см. рисунок к задаче). Программист написал программу с ошибкой.



ПАСКАЛЬ	БЕЙСИК
<pre> Const n=100; Var I,k:integer; A,x,y:real; Begin Randomize; Writeln('Введите сторону треугольника'); Readln(A); Readln(x,y); K:=0; For i:=1 to n do begin X:=-a/2+ random*a*2; Y:=-a/2+ random*a*2; writeln('x=',x:5:2, ' y=',y:5:2); If (x>=0) and (x<=a*sqrt(3) and (y>=0) and (y<=sqrt(3)*(a-x)) then k:=k+1; End; Writeln(k,' точек попало внутрь треугольника '); Readln; end. </pre>	<pre> Const n=100 Defint I,k Randomize Input «Введите сторону треугольника»;A Input «x, y»;x,y K=0; For i=1 to n do X=-a/2+ rnd*2*a Y=-a/2+ rnd*2*a Print «x=»;x;»y=»;y If (x>=0) and (x<=a*sqrt(3) and (y>=0) and (y<=sqrt(3)*(a-x)) then k=k+1 End if End i Print k ;» точек попало внутрь треугольника» end </pre>

- Привести пример таких x, y , для которых программа дает неправильный результат.

О Т В Е Т: $x=a/4, y=a\sqrt{3}/2$. Так как не проверяется ограничение $(y \leq x \cdot \sqrt{3})$, то точка с этими координатами в треугольник не попадает, но она будет учитываться как попавшая, и счетчик k увеличится на единицу. То есть будет неверным общее количество точек, попавших в треугольник, будет выведено большее количество, чем их будет на самом деле.

- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

О Т В Е Т: добавить проверку ограничения $(y \leq x \cdot \sqrt{3})$ со связкой **and** к другим условиям в операторе **If**. Должно быть:

If (x>=0) and (x<=a*sqrt(3)) and (y>=0) and (y<=x*sqrt(3)) and (y<=sqrt(3)*(a-x)) then k:=k+1;

$y=x*\sqrt{3}$ — это уравнение прямой, проходящей через начало координат, и точка должна лежать «ниже» этой прямой.

- Укажите, какая часть программы является лишней.

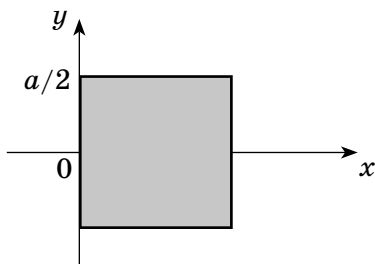
О Т В Е Т: **Readln(x,y);** — лишним является ввод значений переменных x, y, т.к. они должны задаваться с помощью функции генерации случайных чисел. Кроме того, надо вычислить значение квадратного корня из 3 один раз и присвоить его значение переменной, например,

s:= sqrt(3); а далее использовать s в следующем операторе: **If (x>=0) and (x<=a*s) and (y>=0) and (y<=x*s) and (y<=s*(a-x)) then k:=k+1;** То есть лишним является многократное вычисление **sqrt(3)** в операторе

If (x>=0) and (x<=a*sqrt(3)) and (y>=0) and (y<=sqrt(3)*(a-x)) then k:=k+1;

Правильная программа	
ПАСКАЛЬ	БЕЙСИК
<pre>Const n=100; Var I,k:integer; A,x,y,s:real; Begin Randomize; Writeln('Введите сторону треугольника'); Readln(A); K:=0; s:= sqrt(3); For i:=1 to n do begin X:=-a/2+ random*a*2; Y:=-a/2+ random*a*2; writeln('x=',x:5:2, ' y=',y:5:2); If (x>=0) and (x<=a*sqrt(3)) and (y>=0) and (y<=x*sqrt(3)) and (y<=sqrt(3)*(a-x)) then begin k:=k+1; writeln(a*sqrt(3):5:2,x*sqrt(3):5:2,s qrt(3)*(a-x):5:2); writeln('k=',k); end; {use s: If (x>=0) and (x<=a*s) and (y>=0) and (y<=x*s) and (y<=s*(a-x)) then k:=k+1; } End; Writeln(k, ' точек попало внутрь треугольника '); Readln; end.</pre>	<pre>Const n=100 Defint I,k Randomize Input «Введите сторону треуголь- ника »;A K=0; For i=1 to n do X=-a/2+ rnd*2*a Y= -a/2+ rnd*2*a Print «x=«;x;» y=«;y If (x>=0) and (x<=a*sqr(3)) and (y>=0) and (y<=x*sqrt(3)) and (y<=sqr(3)*(a-x)) then k=k+1 End if End i Print k ;« точек попало внутрь треугольника» end</pre>

6. Требовалось написать программу определения количества точек из N точек, сгенерированных случайным образом, попавших в квадрат со стороной a , включая его границы (см. рисунок к задаче). Программист написал программу с ошибкой.



ПАСКАЛЬ	БЕЙСИК
<pre> Const n=100; Var I,k:integer; A,x,y:real; Begin Randomize; Writeln('Введите сторону квадрата'); Readln(A); Writeln('Введите координаты точки x,y'); Readln(x,y); K:=0; For i:=1 to n do begin X:=-a/2+ random*2*a; Y:= -a/2+ random*2*a; Writeln('x=',x:5:2,' y=',y:5:2); If (x>=0) and (x<=a) and (y<=a/2) then k:=k+1; End; Writeln(k,' точек/точки попало внутри квадрата'); Readln; end. </pre>	<pre> Const n=100 Defint I,k Randomize Input «Введите сторону квадрата»;A Input «x, y «;x,y K=0; For i=1 to n do X=-a/2+ RND*2*a Y= -a/2+ RND*2*a Print «x=«;x;»y=«;y If (x>=0) and (x<=a) and (y<=a/2) then k=k+1 End if End i Print k ;« точек/точки попало внутри квадрата» end </pre>

- Привести пример таких координат x, y точки, для которых программа дает неправильный результат.

О Т В Е Т: $x=a/4, y=-a$. Т.к. не проверяется ограничение $(y \geq -a/2)$, то точка с этими координатами в квадрат не попадает, но она будет учитываться как попавшая и счетчик k увеличится на единицу. То есть будет неверным общее количество точек, попавших в квадрат, будет выведено большее количество точек, чем их будет на самом деле.

- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

О Т В Е Т: добавить проверку ограничения $(y \geq -a/2)$ со связкой **and** к другим условиям в операторе **If**. Должно быть:

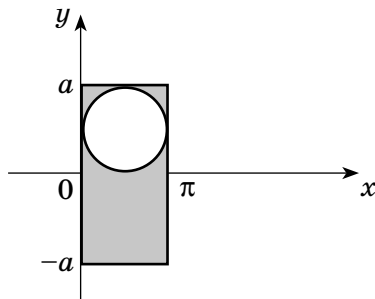
If $(x \geq 0)$ and $(x \leq a)$ and $(y \leq a/2)$ and $(y \geq -a/2)$ then $k:=k+1$;
 $y=-a/2$ — это уравнение прямой, являющейся нижней стороной квадрата, и точка должна лежать выше этой прямой.

- Укажите, какая часть программы является лишней.

О Т В Е Т: Лишним является ввод значений переменных x, y , т.к. они должны задаваться с помощью функции генерации случайных чисел.

Правильная программа	
ПАСКАЛЬ	БЕЙСИК
<pre>Const n=5; Var I,k:integer; A,x,y:real; Begin Randomize; Writeln('Введите сторону квадрата'); Readln(A); K:=0; For i:=1 to n do begin X:=-a/2+ random*2*a; Y:= -a/2+ random*2*a; Writeln('x=',x:5:2,' y=',y:5:2); If (x>=0) and (x<=a) and (y<=a/2) and (y>=-a/2) then k:=k+1; writeln('k=',k); End; Writeln(k,' точек/точки попало внутри квадрата'); Readln; end.</pre>	<pre>Const n=100 Defint I,k Randomize Input «Введите сторону квадрата»;A K=0; For i=1 to n do X=-a/2+ RND*2*a Y= -a/2+ RND*2*a Print «x=«;x;»y=«;y If (x>=0) and (x<=a) and (y<=a/2) and (y>=-a/2) then k=k+1 End if End i Print k ;« точек/точки попало внутри квадрата» end</pre>

7. Требовалось написать программу, чтобы определить, попадет ли точка с координатами (x, y) в заштрихованную область, ограниченную окружностью с центром $(\pi/2, a/2)$ и сторонами прямоугольника $(2a, \pi)$, учитывая границы. Программист написал программу с ошибкой.



ПАСКАЛЬ	БЕЙСИК
<pre> Var a,x,y,z:real; Begin Writeln('введите a, x,y'); Readln(a,x,y); z:=(x-pi/2)*(x-pi/2)+(y-a/2)*(y-a/2); If (x>=0) and (x<=pi) and (y<=a) and (abs(z)>=a*a/4) then writeln('Точка принадлежит области') else writeln('Точка не принадлежит области'); Readln; End.</pre>	<pre> INPUT «Введите a,x, y: «;a,x,y Pi=3.14159265358 z=(x-pi/2)*(x-pi/2)+(y-a/2)*(y-a/2) If (x>=0) and (x<=pi) and (y<=a) and (abs(z)>=a*a/4) then print «Точка принадлежит области» else print «Точка не принадле- жит» End if End</pre>

- Привести пример таких A, x, y , для которых программа дает неправильный результат.

О Т В Е Т: Будет выведено, что точка принадлежит области при $A=1, x=1, y=-2$, но это неверно, т.к. не проверяется условие $y < -a$.

- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

О Т В Е Т: в программе **не проверяется** условие ($y \geq -a$), поэтому его надо добавить с логической связкой **and**.

```

If (x>=0) and (x<=pi) and (y<=a) and (y>=-a) and (z>=a*a/4) then
writeln('принадлежит области') else
writeln(' не принадлежит области');
```

- Укажите, какая часть программы является лишней.

О Т В Е Т: лишним является использование функции вычисления абсолютного значения переменной z : ($abs(z) \geq a*a/4$) в условном операторе **If**, т.к. z является суммой квадратов двух выражений и всегда неотрицательна.

Правильная программа	
ПАСКАЛЬ	БЕЙСИК
<pre> Var a,x,y,z:real; Begin Writeln('введите a, x,y'); Readln(a,x,y); z:=(x-pi/2)*(x-pi/2)+(y-a/2)*(y-a/2); If (x>=0) and (x<=pi) and (y<=a) and (y>=-a) and (z>=a*a/4) then writeln('Точка принадлежит области') else writeln('Точка не принадлежит области'); Readln; End.</pre>	<pre> INPUT «Введите a,x, y: «;a,x,y Pi=3.14159265358 z=(x-pi/2)*(x-pi/2)+(y-a/2)*(y-a/2) If (x>=0) and (x<=pi) and (y<=a) and (y>=-a) and (z>=a*a/4) then print «Точка принадлежит обла- сти» else print «Точка не при- надлежит» End if End</pre>

8. Требовалось написать программу, чтобы прочитать из файла *input.dat* строку и проверить, является ли она записью восьмеричного числа. Вывести сообщение, является ли число восьмеричным. Программист написал программу с ошибкой.

ПАСКАЛЬ	БЕЙСИК
<pre> program file_my; uses crt; Var txt:text; Str:string; {для введенной строки} Ch:char; I,k,n,x,j:integer; Begin Assign(txt,'input.dat'); {Связываем переменную txt с внешним файлом и открываем файл для чтения} Reset(txt); Readln(txt,str); {считываем из файла} Close(txt); {закрываем файл} N:=length(str); j:=0; For i:=1 to n do begin X:=ord(str[i])-ord('0') ; If (x >0) and (x<7) then j:=j+1; end; {конец цикла по i} if j=n then writeln(str, '- число записано в восьмеричной системе счисления') else writeln(str, '- число записано не в восьмеричной системе счисления'); Readln; End. </pre>	<pre> DIM s\$(255) as string Defint I,k,n,x,j Defstr A,B,m,c 'или m\$ A:='input.dat' B:='output.dat' 'Открываем файл A для чтения как 1 OPEN a FOR INPUT AS #1 INPUT\$ #1,m 'm-набор символов. Читает строку символов из файла CLOSE 'закрываем файл N=len(m) For i=1 to n S\$(i)= mid\$(m,I,1) Next i J=0 For i=1 to n X=val(s\$(i)) If (x>0) and (x<7) then j=j+1 Next i If j=n then Print m;«- число записано в восьмеричной системе счисления» else Print m«- число записано не в восьмеричной системе счисления» end </pre>

- Привести пример такой строки, для которой программа дает неправильный результат.

О Т В Е Т: Неправильно для любых строк, содержащих символы 0 (ноль) и 7. Оператор **If (x>0) and (x<7) then j:=j+1;** записан неверно, т.к. цифры 8-й системы счисления — это 0, 1, 2, 3, 4, 5, 6, 7.

Например, строка 103457

Получаем: Число записано **не** в восьмеричной системе счисления.

- Исправьте программу так, чтобы в ней не выполнялось лишних операций.

О Т В Е Т: В программе осуществляются лишние операции из-за преобразования символов в цифры: **X:=ord(str[i])-ord('0');**. Но будет достаточно сравнивать проверяемый символ с символами, которые представляют собой цифры от 0 до 7, т.к. в символы цифр стандарта ASCII располагаются упорядоченно от 0 до 9. Поэтому можно выполнить проверку, находится ли проверяемый символ в диапазоне символов от 0 до 7. Надо заменить цикл


```

For i:=1 to n do begin
  X:=ord(str[i])-ord('0') ;
  If (x >0) and (x<7) then j:=j+1;
end; {конец цикла по i}
на следующий, попутно исправив ошибку, т.е. включить провер-
ку цифр 0 и 7:

```

```

For i:=1 to n do
  If (str[i] >= '0') or (str[i] <= '7') then j:=j+1;

```

• Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

О Т В Е Т: `If (str[i] >= '0') or (str[i] <= '7') then j:=j+1;` Так как 0 и 7 — цифры 8-й системы счисления.

Правильная программа (проверка цифр(чисел) от 0 до 7)	
ПАСКАЛЬ	БЕЙСИК
<pre> program file_my; uses crt; Var txt:text; Str:string; {для введенной строки} Ch:char; I,k,n,x,j:integer; Begin Assign(txt,'input.dat'); {Связыва- ем переменную txt с внешним файлом и открываем файл для чтения} Reset(txt); Readln(txt,str); {считываем из файла} Close(txt); {закрываем файл} N:=length(str); j:=0; For i:=1 to n do begin X:=ord(str[i])-ord('0') ; If (x >=0) and (x<=7) then j:=j+1; end; {конец цикла по i} if j=n then writeln(str, '- число записано в восьмеричной системе счисления') else writeln(str, '- число записано не в восьмеричной системе счисле- ния'); Readln; End. </pre>	<pre> DIM s\$(255) as string Defint I,k,n,x,j Defstr A,B,m,c 'или m\$ A:=«input.dat» B:=«output.dat» 'Открываем файл A для чтения как 1 OPEN a FOR INPUT AS #1 INPUT\$ #1,m 'm-набор симво- лов. Читает строку символов из файла CLOSE 'закрываем файл N=len(m) For i=1 to n S\$(i)= mid\$(m,I,1) Next i J=0 For i=1 to n X=val(s\$(i)) If (x>=0) and (x<=7) then j=j+1 Next i If j=n then Print m;«- число записано в вось- меричной системе счисления» else Print m;«- число записано не в восьмеричной системе счисления» end </pre>
<p>Другой вариант правильной эффективной программы (проверка символов от '0' до '7')</p> <p>В этой программе сразу после обнаружения символа, не являющегося восьмеричным, прекращается проверка последующих символов, т.е. нет лишних операций</p>	

ПАСКАЛЬ	БЕЙСИК
<pre> program file_my; uses crt; Var txt:text; Str:string; {для введенной строки} Ch:char; I,k,n,x,m:integer; Begin Assign(txt,'input.dat'); {Связываем переменную txt с внешним файлом и открываем файл для чтения} Reset(txt); Readln(txt,str); {считываем из файла} Close(txt); {закрываем файл} N:=length(str); For i:=1 to n do If (str[i] < '0') or (str[i] > '7') then begin writeln(str, '- число записано не в восьмеричной системе счисления'); readln; exit; end; writeln(str, '- число записано в восьмеричной системе счисления'); Readln; End.</pre>	<pre> DIM s\$(255) as string Defint I,k,n Defstr A,B,m,c 'или m\$ A:='input.dat' B:='output.dat' 'Открываем файл A для чтения как 1 OPEN a FOR INPUT AS #1 INPUT\$ #1,m 'm-набор символов. Читает строку символов из файла CLOSE 'закрываем файл N=len(m) For i=1 to n S\$(i)= mid\$(m,I,1) Next i For i=1 to n If (s\$(i) < «0») or (s\$(i) > «7») then Print m;«- число записано не в восьмеричной системе счисления» exit End if Next i Print m;«- число записано в восьмеричной системе счисления» end</pre>

9. Требовалось написать программу вычисления **числа сочетаний из n элементов по m**, используя **функцию вычисления факториала**. Программист написал программу с ошибкой.

$$C_n^m = \frac{n!}{m!(n-m)!}$$

ПАСКАЛЬ	БЕЙСИК
<pre> Var c,f1,f2,f3:longint; n,m:integer; {функция вычисления факториала F числа n} {F=1*2*3*...*n, для n=0 и n=1 F=1} Function fact(n:integer):longint; Var K:integer; F:longint; Begin If (n=1) or (n=0) then f:=1 Else begin F:=1; For k:=2 to n do</pre>	<pre> Deflng c,f1,f2,f3 Defint n,m Rem функция вычисления факториала F числа n Rem F=1*2*3*...*n, для n=0 и n=1 F=1 DEF FNfact&(n) If (n=1) or (n=0) then f&=1 Else F&=1 For k=2 to n F&=f&*k FNFact&=f&</pre>

ПАСКАЛЬ	БЕЙСИК
<pre>f:=f*k; Fact:=f; end; End; {основная программа} begin wirte('Введите n='); readln(n); wirte('Введите m<n='); readln(m); f1:=fact(n); f2:=fact(m); f3:=fact(n-m); c:=f1 div (f2*f3); writeln('Число сочетаний из n по m C=',c:6:0); readln; End.</pre>	<pre>End if End i END DEF Input «Введите n=»;n Input «Введите m<n=»;m f1=FNfact&(n) f2=FNfact&(m) f3=FNfact&(n-m) c=f1 \ (f2*f3) print «Число сочетаний из n по m C=';c end</pre>

- Привести пример таких **n** и **m**, для которых программа дает неправильный результат.

О Т В Е Т: Неправильно для **n=1** и **n=0**. Результат сохраняется в локальной переменной **f** и не передается в основную программу.

- Исправьте программу так, чтобы в ней не выполнялось лишних операций.

О Т В Е Т: Лишним является описание переменных **f1,f2,f3** в операторе **Var c,f1,f2,f3:longint;** и их использование в основной программе:

```
f1:=fact(n);
f2:=fact(m);
f3:=fact(n-m);
c:=f1 div (f2*f3);
```

Следует убрать их из описания переменных, удалить вышеприведенные операторы присваивания и заменить их на один:

```
c:= fact(n) div (fact(m) * fact(n-m));
```

- Исправить ошибку в программе так, чтобы она соответствовала поставленной задаче.

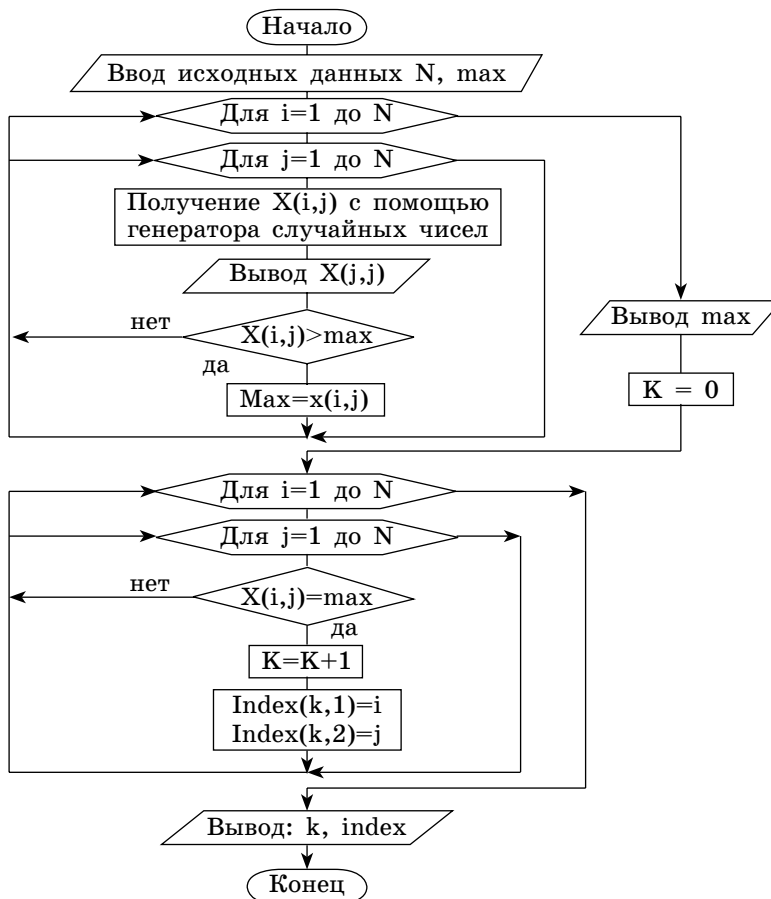
О Т В Е Т: Заменить оператор **If (n=1) or (n=0) then f:=1;** на оператор **If (n=1) or (n=0) then fact:=1;** Значение переменной **fact** передается в основную программу.

Правильная программа	
ПАСКАЛЬ	БЕЙСИК
<pre>Var c:longint; n,m:integer; {*****функция вычисления фак- ториала F числа n *****} {F=1*2*3*...*n, для n=0 и n=1 F=1} Function fact(n:integer):longint;</pre>	<pre>Deflng c Defint n,m Rem функция вычисления факто- риала F числа n Rem F=1*2*3*...*n, для n=0 и n=1 F=1</pre>

ПАСКАЛЬ	БЕЙСИК
<pre> Var K:integer; F:longint; Begin If (n=1) or (n=0) then fact:=1 Else begin F:=1; For k:=2 to n do F:=f*k; Fact:=f; end; End; {*****основная программа*****} begin write('Введите n='); readln(n); write('Введите m<n='); readln(m); c:= fact(n) div (fact(m) * fact(n-m)); writeln('Число сочетаний из n по m C C=',c); readln; End. </pre>	<pre> DEF FNfact&(n) If (n=1) or (n=0) then FNFact&=1 Else F&=1 For k=2 to n F&=f&*k FNFact&=f& End if End i END DEF Rem основная программа Input «Введите n=»;n Input «Введите m<n=»;m c= FNfact&(n) \ (FNfact&(m) * FNfact&(n-m)) print «Число сочетаний из n по m C C=';c end </pre>

Ответы на задания С2

1. Разбор решения задачи 1. Создание блок-схемы алгоритма решения задачи.



Алгоритм решения задачи будет состоять из следующих блоков:

1. Описание двумерного массива любого типа размерности $N \times N$. Описание двумерного массива индексов целого типа размерности $K \times 2$ для хранения пар индексов элементов, равных максимальному.
2. Начало двух вложенных циклов по I и j .
3. Задание массива любым способом (ввод с клавиатуры, с использованием случайной функции) с использованием вложенных циклов. Вывод элементов массива на экран. Так как одновременно с заданием массива можно выполнять поиск максимального элемента, то целесообразно это сделать. Поэтому до цикла следует задать начальное значение переменной для хранения максимального значения, например, $\max = -32768$ (минимальное целое число для типа Integer). Найти максимальный элемент.
4. Конец циклов.
5. Задать начальное значение счетчика количества элементов массива, равных максимальному: $K=0$.
6. Организовать сравнение всех элементов массива с найденным максимальным \max , используя вложенные циклы. Если элемент равен максимальному, то увеличить значение переменной-счетчика на 1 и заполнить соответствующие элементы массива индексов.
7. Вывести значение количества элементов массива, равных максимальному, на экран и их индексы.

Пример правильной программы на языке Паскаль (Turbo Pascal 7.0)

```
Const n=15; {задание константы n=15 — размер массива, задается произвольно}
Type
    mass=array[1..n,1..n] of integer;
    ind=array[1..n,1..2] of integer;
Var
    x:mass;          {описание двумерного массива x типа mass размерности NxN}
    i,j,max,k:integer;
    index:ind; {описание целых переменных и массива индексов}
Begin
    Randomize;
    max:=-32768;    {начальное значение переменной для хранения максимума}
    For i:=1 to n do begin {цикл по I — по строкам}
        For j:=1 to n do begin {цикл по J — по столбцам}
            x[i,j]:=-10+random(40); {получение элементов массива с помощью генератора случайных чисел}
            write(x[i,j]:5);        {вывод элементов массива в строку}
            if x[i,j] > max then max:=x[i,j]; {сравнение текущего элемента с Max,
            если он больше, то max получает его значение}
            end;
            writeln;
        end;
        writeln('max=',max); {вывод значения максимального элемента}
        k:=0; {обнуление счетчика элементов массива, равных максимальному}
        for i:=1 to n do {вложенные циклы}
            for j:=1 to n do
                if x[i,j]=max then begin
                    k:=k+1; {если текущий элемент равен max, то значение счетчика K увеличивается на 1}
```

```

    index[k,1]:=i;
    index[k,2]:=j;
    End;
    writeln('Количество элементов, равных максимальному =',k);
{вывод количества элементов, равных максимальному}
    If k<>0 then begin
        Writeln(' Индексы элементов массива ');
        For i:=1 to k do
            Writeln(index[I,1]:5,index[I,2]:5); {Вывод индексов элементов,
равных максимальному}
        End;
    readln;
end.

```

2. Опишите на русском языке или одном из языков программирования алгоритм перестановки элементов в каждом из столбцов целого двумерного массива A размером $N \times M$ в обратном порядке.

Пример правильной программы на языке Паскаль (Turbo Pascal 7.0)

```

Const n=15;
Type
    mass=array[1..n] of integer;
    mass2=array[1..n,1..n] of integer;
Var
    x:mass; xnn:mass2;
    i,j:integer;
{*****процедура перестановки элементов в одномерном массиве*****}
procedure perest(n:integer;var x:mass);
var y,i:integer;    {буферная переменная y}
Begin
for i:=1 to n div 2 do {заголовок цикла, n div 2 , т.к. количество
перестановок
в 2 раза меньше количества элементов, div — деление нацело.
При нечетном количестве элементов средний элемент переставлять не
с чем.}
begin
y:=x[i];    {запоминание в переменной y текущего элемента массива}
x[i]:=x[n-i+1];    {текущий элемент получает значение симметричного
элемента с конца массива}
x[n-i+1]:=y;    {элемент с конца массива получает значение симме-
тричного элемента с начала массива}
end;    {конец цикла}
end; {the end of procedure}
{*****начало основной программы *****}
Begin
Randomize;
Writeln('Исходный массив');
For i:=1 to n do begin
    For j:=1 to n do begin
        xnn[I,j]:=-10+random(40); {генерирование массива целого типа}
        write(xnn[I,j]:5);
    end;
end;
writeln;
end;

```

```

For j:=1 to n do begin
  For i:=1 to n do
    X[i]:=xnn[i,j];
    Perest(n,x);           {Обращение к процедуре}
    For i:=1 to n do
      xnn[i,j]:=X[i];
end;
writeln('Результат перестановки элементов');
For i:=1 to n do begin
  For j:=1 to n do
    write(xnn[i,j]:5); {вывод столбца массива X с переставленными эле-
ментами}
    writeln;
end;
readln;
end.

```

Примечание: то же для массива любого типа, изменить только описание типа

3. Опишите на русском языке или одном из языков программирования алгоритм определения количества элементов столбца двумерного массива A размером $N \times N$, равных максимальному, в каждом столбце. Поместите их в отдельный массив.

Пример правильной программы на языке Паскаль (Turbo Pascal 7.0)

```

Const n=15;
Type
  mass=array[1..n,1..n] of integer; {исходный массив}
  mass1= array[1..n] of integer;    {массив результата}
Var
  x:mass;x1,y:mass1; {описание двумерного X и одномерных Y,
X1 массивов}
  i,j,k:integer;    {описание целых переменных}
Begin
  Randomize;
  For i:=1 to n do begin
    x1[i]:=-32768; {задание начальных значений элементам
массива X1.
-32768 - минимальное целое число для переменных типа Integer }
    For j:=1 to n do
      begin
        x[j,i]:=-10+random(40); {получение исходного массива с
помощью генератора случайных чисел}
        write(x[j,i]:5); {вывод исходного массива}
        if x[j,i] > x1[i] then x1[i]:=x[j,i]; {поиск макси-
мальных элементов в столбцах, заполнение массива X1 }
      end;
      writeln;
    end; {конец цикла i}
  writeln('Максимальные элементы в столбцах');
  for i:=1 to n do
  writeln(i,x1[i]:5); {вывод максимальных элементов в столбцах}
  writeln;writeln('Количество максимальных элементов в столбцах');

```

```

for i:=1 to n do begin
  k:=0;
  for j:=1 to n do
    if x[j,i]=x1[i] then k:=k+1; {если текущий элемент столбца равен
    максимальному в данном столбце, то значение счетчика К увеличивается на 1}
    y[i]:=k;
    writeln(I,k:5); {вывод количества элементов равных максимальному в столбце i}
  end;
end;
readln;
end.

```

4. Массив содержит несколько предложений, разделенных **точкой, вопросительным или восклицательным знаками**. Опишите на русском языке или одном из языков программирования алгоритм вывода на экран **количества предложений**.

Пример правильной программы на языке Паскаль (Turbo Pascal 7.0)

```

Var a:string; {описание символьной переменной}
I,k:integer; {описание целых переменных}
Begin
Write('a=');readln(a); {ввод предложений}
K:=0; {обнуление счетчика количества предложений}
For i:=1 to length(a) do {организация цикла, конечное значение
переменной цикла I равно количеству символов во введенных предло-
жениях,
используется функция length }
if (a[i]='.') or (a[i]='!') or (a[i]='?') then k:=k+1; {увеличение K на
единицу}
writeln('Количество предложений k=',k:5);
Readln;
End.

```

5. Опишите на русском языке или одном из языков программирования алгоритм определения суммы элементов в строках массива **A** размером **N×M** и его столбцах. Сохраните эти суммы в одном массиве: сначала суммы строк, затем суммы столбцов.

Пример правильной программы на языке Паскаль (Turbo Pascal 7.0)

```

Uses crt;
Const n=7; m=5;
Type
  mass=array[1..n,1..m] of integer;
  mass1=array[1..n+m] of integer;
Var
  x:mass;y:mass1;
  i,j,k,s:integer;
  txt: file of integer;
Begin
{Ввод массива с клавиатуры*}
For i:=1 to n do
For j:=1 to m do

```



```

begin
{writeln('Введите элементы массива по строкам');
readln(x[i,j]);
end;
{Ввод массива из input.dat — внешнего файла*}
Assign(txt,'input.dat'); {Связываем переменную txt с внешним файлом
и открываем файл для чтения}
Reset(txt);
For i:=1 to n do
For j:=1 to m do
Read(txt,x[i,j]); {считываем данные из файла}
Close(txt); {закрываем файл}
{Вывод элементов массива на экран}
For i:=1 to n do begin
For j:=1 to m do
write(x[I,j]:5);
writeln;
end;
{Поиск сумм элементов массива по строкам}
For i:=1 to n do begin
S:=0;
For j:=1 to m do
S:=s+x[I,j];
Y[i]:=s;
End;
K:=n; {нумерация элементов массива должна быть продолжена}
{Поиск сумм элементов массива по столбцам}
For i:=1 to m do begin
S:=0;
For j:=1 to n do
S:=s+x[j,i];
K:=k+1;
Y[k]:=s;
End;
For i:=1 to m+n do {можно использовать k, т.к. k=m+n}
Write(y[i]:5); {вывод массива сумм по строкам и столбцам}
readln;
end.

```

* — массив в программе вводится только одним из приведенных способов на выбор

6. Опишите на русском языке или одном из языков программирования алгоритм поиска элементов целого массива X размером N , являющихся простыми числами. Определите их количество и сохраните в отдельном массиве. Элементы массива имеют значения от 0 до 50.

Пример правильной программы на языке Паскаль (Turbo Pascal 7.0)

```

Const n=15;
Type
    mass=array[1..n] of integer;
Var
    X,y:mass;    {описание массивов x — исходный, y — результат}

```

```

i,k,m:integer;           {описание переменных целого типа}
{*****Процедура определения простого числа*****}
procedure prost(a:integer;var k:integer);
var i:integer;
begin
K:=0;
For i:=2 to a div 2 do
If (a mod i)=0 then k:=k+1;
End;
{*****Начало основной программы*****}
Begin
Randomize;
For i:=1 to n do
begin
x[i]:=-10+random(40); {генерирование одномерного массива цело-
го типа}
write(x[i]:5);           {вывод элементов массива на экран}
end;                     {end i}
writeln;
m:=0;                    {Обнуление счетчика простых чисел в массиве}
for i:=1 to n do        {Заголовок цикла}
begin
prost(x[i],k);          {обращение к процедуре}
if k=0 then begin      {Если k=0, то число простое}
m:=m+1;                {увеличение счетчика количества элементов,
удовлетворяющих условию, на 1}
y[m]:=x[i];            {заполнение массива у простым числом}
end; {конец IF}
end; {конец цикла}
writeln('кол-во простых чисел в массиве X: m=',m); {вывод количества
элементов, удовлетворяющих условию}
If m> 0 then begin
for i:=1 to m do
write(y[i]:6);          {вывод массива простых чисел, если m>0}
end; {end if}
readln;
end.

```

Ответы на задания С4

1. Алгоритм решения задачи.

1. Для представления единой структурой совокупности неоднородных данных (название — символы, количество продукции (месяц1, месяц2, месяц3) — целые числа 4 байта) будем использовать записи (record). Для ссылки на поле записи используется форма: <имя переменной типа запись>.<имя поля>.

р:массив[1..100] = запись

имя: символьный тип;

продукция: массив[1..3] целый тип; {3 месяца};

2. Ввод количества филиалов n.

3. Начало цикла с параметром i от 1 до n для чтения полей записей.

а. заполнение nz — массива индексов записей.

- b. Переменной name присвоить пустой символ.
 - c. Повторять {цикл с постусловием для считывания названия филиала}.
 - d. Ввод символа.
 - e. Накапливание символов фамилии в переменной name.
 - f. Пока символ не будет пробелом.
 - g. Обнуление i-того элемента массива ms для подсчета суммы произведенной продукции для данного филиала.
 - h. Начало цикла по j от 1 до 3 (3 месяца) для считывания количества продукции.
 - i. Считывание количества текущего месяца.
 - ii. Накапливание количества продукции в ms[j].
 - i. Конец цикла по j.
 - j. Вывод суммарной продукции ms[i].
 - k. Конец цикла по i — считаны все записи, выведен и заполнен массив ms.
4. Упорядочение массива ms по убыванию методом обмена («метод пузырька»).
 5. Вывод упорядоченного массива ms.
 6. Вывод лучших филиалов.
 - Задание начальных значений для цикла While (k=1).
 - Пока k<=2 делать.
 - ✓ Вывод филиалов с максимальной суммарной продукцией.
 - ✓ Вывод филиалов с суммарной продукцией следующей за максимальной.
 - Конец цикла.
 7. Вывод филиалов-аутсайдеров (см. п. 6), но с минимальной суммарной продукцией и с суммарной продукцией, предшествующей минимальной.
 8. Конец программы.

**Пример правильной и эффективной программы на языке Паскаль
(Borland Turbo Pascal 7.0)**

```

var p:array[1..50] of record
    name: string[10];
    m:array[1..3] of longint;
end;

c:char;
i,i1,n,j,k,imax:integer;
max,dif:longint;
ms: array[1..50] of longint;
nz: array[1..50] of byte;           {массив индексов записей}
begin
readln(n);                         {Ввод количества филиалов}
for i:=1 to n do
begin
  nz[i]:=i;                         {заполнение массива индексов записей}
  p[i].name:='';
  repeat
    read(c);
    p[i].name:=p[i].name+c
  until c=' ';                       {считано название филиала}
  ms[i]:=0;                          {обнулено кол-во продукции за квартал}
  for j:=1 to 3 do begin              {3 месяца}
    read(p[i].m[j]);                 {считывание кол-ва продукции по месяцам}
  
```

```

ms[i]:= ms[i] +p[i].m[j];
end;                                     {цикла j}
readln;
writeln('i=',i, ' ms=',ms[i]);         {вывод суммарной продукции ms}
End;                                     {цикла i}
{упорядочение массива ms по убыванию}
{*****}
For i:=1 to n-1 do
begin
Max:=ms[i]; i1:=nz[i];
Imax:=I;
For j:=i+1 to n do
begin
If ms[j]>max then begin
max:=ms[j];i1:=nz[j];
imax:=j;
end;                                     {end if}
end;                                     {end j}
ms[imax]:=ms[i];nz[imax]:=nz[i];
ms[i]:=max; nz[i]:=i1;
writeln(i:5,nz[i]:5,ms[i]:5);
End;                                     {end i}
{*****}
for i:=1 to n do begin i1:=nz[i]; {вывод упорядоченного массива ms}
writeln(i:5,p[i1].name:5,ms[i]:10); end;
{вывод лучших филиалов и филиалов-аутсайдеров}
I:=1;k:=1;max:=ms[1]; writeln('Лучшие филиалы:');
While k<=2 do {вывод лучших }
begin
i1:=nz[i];
if max=ms[i] then with p[i1] do writeln(name,ms[i])
else
if (k<2) then begin
max:=ms[i];k:=k+1; {2-й лучший результат}
with p[i1] do writeln(name,ms[i]);
end {end if}
else k:=3;
i:=i+1;
end;                                     {end while}
writeln;
k:=1;max:=ms[n];i:=n; writeln('Филиалы-футсайдеры:');
while k<=2 do
begin
i1:=nz[i];
if max=ms[i] then with p[i1] do writeln(name,ms[i])
else
if (k<2) then begin
max:=ms[i];k:=k+1; {2-й худший результат}
with p[i1] do writeln(name,ms[i]);
end {end if}
else k:=3;
i:=i-1;
end;                                     {end while}
writeln;
dif:=ms[1]-ms[n];
writeln(' Разница Max-Min=',dif);
writeln;
Readln;
End.

```

2. На вход программе подаются сведения об учениках школы (не более 300 чел.). В первой строке сообщается количество учащихся N , каждая из следующих N строк имеет формат: <Фамилия> <Имя> <Отчество> <Пол>, <Год рождения>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 12 символов, <Отчество> строка, состоящая не более чем из 12 символов, <Пол> — 1 символ (например, м или ж), <Год рождения> — целое четырехзначное число. <Фамилия> <Имя> <Отчество> <Пол>, <Год рождения> разделены одним пробелом.

Пример входных строк:

150

Кузнецов Сергей Иванович м 1995
Светлов Андрей Петрович м 1997
Андреева Анна Сергеевна ж 1996
Кузнецов Сергей Дмитриевич м 1995
Петрова Елена Антоновна ж 1998

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая выведет отдельно списки девочек и мальчиков в **порядке возрастания их года рождения**.

Пример выходных строк:

Девочки

Андреева Анна Сергеевна 1996
Петрова Елена Антоновна 1998

Мальчики

Кузнецов Сергей Иванович 1995
Кузнецов Сергей Дмитриевич 1995
Светлов Андрей Петрович 1997

О Т В Е Т: *Пояснение:* В данной задаче выполнена сортировка по возрастанию самих записей. В *следующей задаче* введен двумерный массив **sum**, в первом столбце которого хранится суммарный балл абитуриента, а во втором номер записи (можно наоборот). При упорядочении массива по убыванию во второй столбец записываются номера записей, которые потом используются при выводе результата. То есть порядок записей не меняется.

Пример правильной и эффективной программы на языке Паскаль (Borland Turbo Pascal 7.0)

```
var p:array[1..300] of record
    famm:string[44];
    sex:char;
    year:integer;
end;
c,xsex:char;
I,j,n,imin,min,k:integer;
```

```

fmin:string[44];
begin
xreadln(n); {количество уч-ся}
for i:=1 to n do
  begin
    p[i].famm:='';
    repeat
      read(c);
      p[i].famm:=p[i].famm+c
    until c=' '; {считана фамилия}
    repeat
      read(c);
      p[i].famm:=p[i].famm+c
    until c=' '; {считано имя}
    repeat
      read(c);
      p[i].famm:=p[i].famm+c
    until c=' '; {считано отчество}
    read(p[i].sex,c); {считан пол} {считан пробел}
    read(p[i].year); {считан год}
    readln; {конец i-той записи}
  end;
{Упорядочить по возрастанию весь список}
For i:=1 to n-1 do begin
Min:=p[i].year;
Imin:=I;
k:=0;
For j:=i+1 to n do
  If p[j].year < min then begin
    k:=1; {признак существ. мин. элемента }
    min:=p[j].year;
    fmin:=p[j].famm;
    xsex:=p[j].sex;
    imin:=j;
  end; {end if}
if k=1 then begin
p[imin].year:=p[i].year;
p[i].year:=min;
p[imin].famm:=p[i].famm;
p[i].famm:=fmin;
p[imin].sex:=p[i].sex;
p[i].sex:=xsex;
end; {end if}
End; {end i}
{Вывод результата}
Writeln('Девочки');
for i:=1 to n do
with p[i] do begin
if sex='f' then writeln(famm,year:5);
end; {конец цикла i}
Writeln('Мальчики');
for i:=1 to n do
with p[i] do begin
if sex='m' then writeln(famm,year:5);
end; {конец цикла i}
readln;
end.

```

3. На вход программе подаются сведения об абитуриентах некоторого вуза. В первой строке сообщается количество абитуриентов N , которое больше 20, но меньше или равно 100, каждая из следующих N строк имеет следующий формат: <Фамилия> <Имя> <Отчество> <Балл1> <Балл 2> <Балл 3>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 15 символов, <Отчество> — строка, состоящая не более чем из 20 символов, <Балл1> <Балл 2> <Балл 3> — целое число от 0 до 100. <Фамилия>, <Имя> и <Отчество>, а также <Балл1> <Балл 2> <Балл 3> разделены одним пробелом. <Отчество> и <Балл1> разделены двоеточием. Входные данные не упорядочены никаким способом.

Пример входных строк:

```
45
Федорова Ирина Евгеньевна: 80 56 76
Иванов Сергей Захарович: 59 77 90
Светлов Андрей Иванович: 89 67 80
```

Требуется написать программу, которая определит сумму баллов абитуриентов и будет выводить на экран их фамилии, имена, отчества и сумму баллов в порядке убывания. Список пронумеровать.

Пример выходных строк:

1. Светлов Андрей Иванович: 236
2. Иванов Сергей Захарович: 226
3. Федорова Ирина Евгеньевна: 212

О Т В Е Т: *Пояснение:* В данной задаче в отличие от *предыдущей* введен двумерный массив `sum`, в первом столбце которого хранится суммарный балл абитуриента, а во втором — номер записи (можно наоборот). При упорядочении массива по убыванию во второй столбец записываются номера записей, которые потом используются при выводе результата. В *предыдущей* задаче выполнена сортировка самих записей. Использование вспомогательного массива (можно одномерного, см. следующие задачи, массив `nz`) является предпочтительным.

Пример правильной и эффективной программы на языке Паскаль (Borland Turbo Pascal 7.0)	
<pre>var a: array[1..100] of record fio:string[55]; b1:integer; b2:integer; b3:integer; end; sum:array[1..100,1..2] of integer; c:char; i,j,k,nz,n,m,im: integer; begin readln(n);</pre>	<pre>{ФИО абитуриентов} {Балл1 абитуриента} {Балл2 абитуриента } {Балл3 абитуриента } {массив для хранения суммарных баллов и номера записи}</pre>
<pre> readln(n); {считывается количество записей}</pre>	

```

for i:=1 to n do begin
sum[i,1]:=0;           {обнуление суммарных баллов}
sum[I,2]:=I;          {Нумерация записей}
end;
for i:=1 to N do
  with a[i] do
  begin
  fio:='';
  repeat
    read(c);
    fio:=fio+c
  until c='.'; {читаны ФИО}
  read(b1); {читан 1 балл}
  read(b2); { считан 2 балл }
  read(b3); { считан 3 балл }
  readln;
  sum[i,1]:=b1+b2+b3;
  end;
  {Упорядочение массива по убыванию}
For i:=1 to n-1 do begin
M:= sum [I,1];
Im:=I;k:=0;
For j:=i+1 to n do
  If sum [j,1]>m then begin
    m:= sum [j,1];nz:=sum[j,2];
    im:=j; k:=1;
  end; {end if}
if k>0 then begin
sum [im,1]:= sum [I,1];
sum [I,1]:=m;
sum[im,2]:=sum[i,2];
sum[I,2]:=nz;
end; {end if}
End; {end i}
  writeln('Список абитуриентов');
  {цикл для вывода абитуриентов по убыванию баллов}
  for i:=1 to n do begin
    nz:=sum[I,2]; {nz — номер записи}
  {Вывод номера в списке, ФИО и суммарного балла}
    writeln(i,' ',a[nz].fio,' ',sum[I,1]);
  end; {end i}
  readln;
end.

```

4. На вход программе подаются сведения о пациентах больницы одного года. В первой строке сообщается количество пациентов N , которых больше 10, но не более 300, каждая из следующих N строк имеет следующий формат: <Фамилия> <Имя> <Отчество> <Номер истории болезни> <Пол> <Год рождения> <Дата поступления> <Дата выписки>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 15 символов, <Отчество> — строка, состоящая не более чем из 15 символов, <Номер истории болезни> — целое

число, <Пол> — буква “м” или “ж”, <Год рождения> — целое число, <Дата поступления> <Дата выписки> имеют формат чч.мм, где чч — число, мм — номер месяца разделены точкой (например, 04.12). <Фамилия>, <Имя>, <Отчество>, <Номер истории болезни>, <Пол>, <Год рождения>, <Дата начала болезни>, <Дата окончания болезни> разделены одним пробелом.

Примечание: январь, март, май, июль, август, октябрь и декабрь содержат 31 день, февраль — 28, остальные месяцы — 30 дней.

Пример входной строки:

```
200
Федорова Ирина Павловна 345 ж 1967 04.10 20.10
Иванов Сергей Петрович 1029 м 1940 12.12 23.12
Петров Сергей Иванович 4567 м 1993 30.04 10.05
Волохова Зинаида Евгеньевна 345 ж 1992 30.06 10.08
```

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая определит, сколько дней находился в больнице каждый из пациентов.

Вывести номер истории болезни, фамилию, инициалы пациентов и количество дней. **Список вывести в алфавитном порядке.** Исходные данные вводятся корректно.

Результат работы программы может выглядеть так:

Номер ИБ	Фамилия, И.О.	Количество дней
1234	Волохова З.Е.	11
2345	Иванов С.П.	12
123	Петров С.И.	11
222	Федорова И.П.	42

```

Пример правильной и эффективной программы на языке Паскаль
(Borland Turbo Pascal 7.0)

Const d:array[1..12] of integer = (31,28,31,30,31,30,31,31,30,31,30,31);
var p:array[1..300] of record
    name: string[50];
    kp:integer;
    pol: char;
    god: integer;
    data1:string[5];
    data2:string[5];
end;
days,nz:array[1..300] of integer;
{days – массив для хранения времени госпитализации}
{nz – массив для хранения номера записи}
fio1: array[1..300] of string[27]; {массив хранения ФИО с инициалами}
c,cd:char; fio:string[27]; {вспомогательные переменные}
d1,d2,m1,m2,s1,s2,i,k,n,m,j:integer;
begin
    readln(n); {ввод количества записей}

```

```

for i:=1 to n do
with p[i] do begin
  fio:=''; name:='';
  repeat
    read(c);name:=name+c;
    fio:= fio +c
  until c=' '; {считана фамилия}
  read(c); name:=name+c;  fio:= fio +c+'. '; {добавление 1-го инициала
и точки}
  repeat
    read(c); name:=name+c;
    until c=' '; {считано имя}
  read(c); name:=name+c;
  fio:= fio +c+'. '; {добавление 2-го инициала и точки}
  {fio – фамилия с инициалами}
  repeat
    read(c); name:=name+c;
    until c=' '; {считано отчество}
  fio1[i]:=fio; {запомнили фамилию и инициалы}
  read(kp); read(c); {считан номер истории болезни (ИБ) }{считан пробел}
  read(c);pol:=c; read(c); {считан пол м или ж} {считан пробел}
  read(god); read(c); {считан год рождения} {считан пробел}
  read(c,cd); {считан день поступления}
  d1:=(ord(c)-ord('0'))*10 +ord(cd)-ord('0');
  read(c,m1); {пропущена точка и считан месяц поступления}
  read(c,c,cd); {считан пробел и считан день выписки}
  d2:=(ord(c)-ord('0'))*10 +ord(cd)-ord('0');
  readln(c,m2); {пропущена точка и считан месяц выписки}
  Nz[i]:=1; {заполнение массива индексов записей}
  {определение разницы между двумя датами}
  If m1=m2 then
    days[i]:=d2-d1+1 {госпитализация в одном месяце}
  Else if m2=m1+1 then Days[i]:=d2+(d[m1]- d1)+1 {госпитализация
в соседних месяцах}
  else begin
    s1:=0;s2:=0;
    for j:=1 to m1-1 do s1:=s1+d[j];
    for j:=1 to m2-1 do s2:=s2+d[j];
    Days[i]:=s2+d2-(s1+d1)+1; {госпитализация в течение нескольких
месяцев}
  end; {end if else}
End; {конец цикла i}
{упорядочение ФИО по алфавиту с учетом массива индексов записей}
{сортировка методом обмена}
For i:=1 to n-1 do
  For j:=i+1 to n do
    If fio1[i]>fio1[j] then
      begin
        fio:=fio1[i];k:=nz[i];
        fio1[i]:= fio1[j];nz[i]:=nz[j];
        fio1[j]:=fio;nz[j]:=k;
      End; {end if}
  {вывод фамилий пациентов и сроков госпитализации}
  Writeln('Номер ИБ Фамилия И.О. Срок госпитализации');
For i:=1 to n do begin
  K:=nz[i];
  Writeln(p[k].kp:8,' ',fio1[i], days[k]:12);
end;
Readln;
end.

```

5. На вход программе подаются сведения об учениках школы (не более 300 чел.) В первой строке сообщается количество учащихся N, каждая из следующих N строк имеет формат: <Фамилия> <Имя> <название класса><Физкультурная группа>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 12-ти символов, <номер класса> — целое число (например, 11, 9, 10), <Физкультурная группа> — целое число (1,2 или 3), <Фамилия>, <Имя>, <Номер класса>, <Физкультурная группа> — разделены одним пробелом.

Пример входных строк:

```
150
Кузнецов Сергей 9 3
Светлов Андрей 11 2
Платонов Сергей 9 1
Сидоров Юрий 11 1
Петрова Елена 10 2
Иванов Дмитрий 9 3
```

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая определит количество учащихся в каждой группе и выведет **список учащихся по физкультурным группам** начиная с 1-й. Списки пронумеровать, алфавитный порядок в группах можно не соблюдать.

Пример выходных строк:

```
1 физкультурная группа — 2 уч-ся.
Платонов Сергей 9 кл.
Сидоров Юрий 11 кл.

2 физкультурная группа — 2 уч-ся.
Светлов Андрей 11кл.
Петрова Елена 10 кл.

3 физкультурная группа — 2 уч-ся.
Кузнецов Сергей 9 кл.
Иванов Дмитрий 9 кл.
```

Пример правильной и эффективной программы на языке Паскаль (Borland Turbo Pascal 7.0)	
<pre>var p:array[1..300] of record famm:string[20]; name: string[12]; klass,fizra: integer; end; c:char; I,j,n,k,k1,k2,k3:integer; Fm:string; F1,f2,f3: array[1..300] of byte;</pre>	<pre>{короткое целое без знака}</pre>

```

begin
readln(n);    {количество уч-ся}
k1:=0;k2:=0;k3:=0;
for i:=1 to n do
With p[i] do
Begin
  famm:='';
  repeat
    read(c);
    famm:=famm+c
  until c=' '; {считана фамилия}
{fam[i]:=famm; }
  name:='';
  repeat
    read(c);
    name:=name+c
  until c=' '; {считано имя}
  read(klass); {считан класс}
  readln(fizra); {считана физгруппа}
{заполнение массивов индексами для каждой из физгрупп}
Case fizra of
1 : begin k1:=k1+1;f1[k1]:=i;end;
2 : begin k2:=k2+1;f2[k2]:=i;end;
3 : begin k3:=k3+1;f3[k3]:=i;end
End;    {end Case}
end; {end i}
if k1<>0 then begin {проверка, т.к. неизвестно, есть ли уч-ся во всех гр.}
writeln('1 физкультурная группа - ',k1:5,' уч-ся. ');
for i:=1 to k1 do begin
j:=f1[i];
with p[j] do writeln(I, '.', famm,name,klass, ' кл. ');
end;    {end j}
end ;    {end if}
if k2<>0 then begin {проверка, т.к. неизвестно, есть ли уч-ся во всех гр.}
writeln('2 физкультурная группа - ',k2:5,' уч-ся. ');
for i:=1 to k2 do begin
j:=f2[i];
with p[j] do writeln(I, '.', famm,name,klass, ' кл. ');
end;    {end j}
end ;    {end if}
if k3<>0 then begin {проверка, т.к. неизвестно, есть ли уч-ся во всех гр.}
writeln('3 физкультурная группа - ',k3:5,' уч-ся. ');
for i:=1 to k3 do begin
j:=f3[i];
with p[j] do writeln(I, '.', famm,name,klass, ' кл. ');
end;    {end j}
end ;    {end if}
readln;
end.

```

6. На вход программе подаются сведения об учениках школы (не более 300 чел.) В первой строке сообщается количество учащихся N, каждая из следующих N строк имеет формат: <Фамилия> <Имя> < класс> <Оценка1> <Оценка2 >, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 12 символов, < класс> — целое

число (например, 11, 9, 10), <Оценка1> <Оценка2 > — целое число от 1 до 5. <Фамилия>, <Имя>, <класс>, <Оценка1>, <Оценка2 > разделены одним пробелом.

Пример входных строк:

```
150
Кузнецов Сергей 9 5 4
Светлов Андрей 10 5 3
Троекурова Маша 11 3 3
Андреев Сергей 9 4 3
Платов Иван 11 3 4
Денисов Олег 10 3 3
```

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая выведет **список учащихся, упорядоченный по убыванию суммы баллов. Трех лучших** отметить словом **лидеры**. Если среди учащихся будут те, которые набрали ту же сумму баллов, что и лидеры, то отметить и их фамилии.

Пример выходных строк:

Фамилия имя	Класс	Сумма баллов
Кузнецов Сергей	9	9 лидер
Светлов Андрей	10	8 лидер
Андреев Сергей	9	7 лидер
Платов Иван	11	7 лидер
Троекурова Маша	11	6
Денисов Олег	10	6

Пример правильной и эффективной программы на языке Паскаль (Borland Turbo Pascal 7.0)	
<pre>var p:array[1..300] of record fam:string[20]; name: string[12]; klass,ball1,ball2,s: integer; end; c,m:char; I,j,n,i1,b1,max,k:integer; Nz: array[1..300] of byte; begin readln(n); {количество учащихся} for i:=1 to n do With p[i] do Begin fam:=''; repeat read(c); fam:=fam+c until c=' '; {считана фамилия} name:=''; repeat</pre>	

```

    read(c);
    name:=name+c
    until c=' ';           {считано имя}
readln(klass,ball1,ball2); {считаны класс и оценки}
nz[i]:=I; {заполнение массива индексов записей}
s:=ball1+ball2;
end;           {конец цикла i}
{Упорядочение по возрастанию суммы баллов с учетом номеров записей}
{Сортировка методом обмена}
For i:=1 to n-1 do
  For j:=i+1 to n do
    If p[i].s<p[j].s then
      begin
        b1:=p[i].s;i1:=nz[i];
        p[i].s:= p[j].s;nz[i]:=nz[j];
        p[j].s:=b1;nz[j]:=i1;
      End;   {end if}
    {вывод упорядоченного массива фамилий и суммы баллов + лидер для
    победителей}
    writeln('Фамилия имя Класс Сумма баллов'); {Вывод заголовка таблицы}
    max:=p[1].s; {max – первый лучший результат}
    k:=1; {k – счетчик трех лучших результатов по сумме баллов }
    for i:=1 to n do begin
      i1:=nz[i]; {получение номера записи}
      if max=p[i].s then with p[i1] do writeln(fam,name,klass:5,p[i].s:7,' -
      lider')
        else
          if (k<3) then begin
            max:=p[i].s;k:=k+1; {2-й и 3-й лучшие результаты}
            with p[i1] do writeln(fam,name,klass:5,p[i].s:7,' - lider');
          end
          else
            with p[i1] do writeln(fam,name,klass:5,p[i].s:7);
        end; {конец цикла i}
      readln;
    end.

```

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. *Андреева Е.В., Лещинер В.Р., Самылкина Н.Н., Якушкин П.А., Крылов С.С.* Методические рекомендации по оцениванию заданий с развернутыми ответами. Информатика. ЕГЭ 2009. — М.: Федеральный институт педагогических измерений, 2009.
2. *Бешенков С.А., Ракитина Е.А.* Информатика. Систематический курс: Учебник для 10 класса. — М.: Лаборатория Базовых Знаний, 2001. — 432 с.: ил.
3. *Лихтарников Л.М.* Занимательные логические задачи. — СПб.: Лань, МИК, 1996. — 126 с.
4. *Самылкина Н.Н.* Построение тестовых заданий по информатике: Методическое пособие. — М.: БИНОМ. Лаборатория знаний, 2003. — 176 с.: ил.
5. *Самылкина Н.Н., Русаков С.В., Шестаков А.П., Баданина С.В.* Готовимся к ЕГЭ по информатике: Элективный курс: учебное пособие. — М.: БИНОМ. Лаборатория знаний, 2008. — 298 с. с ил.
6. *Смаллиан Р.* Принцесса или тигр?: Пер. с англ. / Под ред. и с предисл. Ю.И. Манина. — М.: Мир, 1985. — 221 с.
7. www.ege.edu.ru/ Демонстрационный вариант по информатике за 2005—2010 гг.

СОДЕРЖАНИЕ

<i>Введение</i>	3
ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ	7
ОСНОВЫ ЛОГИКИ	12
ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ. МОДЕЛИРОВАНИЕ И КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ	18
АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ	27
<i>Ответы</i>	57
<i>Список использованной литературы</i>	94

Издание для дополнительного образования

Для старшего школьного возраста

ЕГЭ. ТЕМАТИЧЕСКИЕ ТРЕНИРОВОЧНЫЕ ЗАДАНИЯ

**Самылкина Надежда Николаевна
Островская Екатерина Михайловна**

ЕГЭ 2013

Информатика

Тематические тренировочные задания

Ответственный редактор *А. Жилинская*
Ведущий редактор *Т. Судакова*
Художественный редактор *Е. Брынчик*
Технический редактор *Л. Зотова*
Компьютерная верстка *А. Попов*

ООО «Издательство «Эксмо»
127299, Москва, ул. Клары Цеткин, д. 18/5. Тел. 411-68-86, 956-39-21.
Home page: www.eksmo.ru E-mail: info@eksmo.ru

Оптовая торговля книгами «Эксмо»:
ООО «ТД «Эксмо». 142700, Московская обл., Ленинский р-н, г. Видное,
Белокаменное ш., д. 1, многоканальный тел. 411-50-74.
E-mail: reception@eksmo-sale.ru

**По вопросам приобретения книг «Эксмо» зарубежными оптовыми
покупателями обращаться в отдел зарубежных продаж ТД «Эксмо»**
E-mail: international@eksmo-sale.ru

International Sales: International wholesale customers should contact
Foreign Sales Department of Trading House «Eksmo» for their orders.
international@eksmo-sale.ru

**По вопросам заказа книг корпоративным клиентам, в том числе в специальном
оформлении, обращаться по тел. 411-68-59, доб. 2299, 2205, 2239, 1251.**
E-mail: vipzakaz@eksmo.ru

**Оптовая торговля бумажно-беловыми
и канцелярскими товарами для школы и офиса «Канц-Эксмо»:**
Компания «Канц-Эксмо»: 142702, Московская обл., Ленинский р-н, г. Видное-2,
Белокаменное ш., д. 1, а/я 5. Тел./факс +7 (495) 745-28-87 (многоканальный).
e-mail: kanc@eksmo-sale.ru, сайт: www.kanc-eksmo.ru

Полный ассортимент книг издательства «Эксмо» для оптовых покупателей:
В Санкт-Петербурге: ООО СЗКО, пр-т Обуховской Обороны, д. 84Е. Тел. (812) 365-46-03/04.
В Нижнем Новгороде: ООО ТД «Эксмо НН», ул. Маршала Воронова, д. 3. Тел. (8312) 72-36-70.
В Казани: Филиал ООО «РДЦ-Самара», ул. Фрезерная, д. 5. Тел. (843) 570-40-45/46.
В Ростове-на-Дону: ООО «РДЦ-Ростов», пр. Стачки, д. 243А. Тел. (863) 220-19-34.
В Самаре: ООО «РДЦ-Самара», пр-т Кирова, д. 75/1, литера «Е». Тел. (846) 269-66-70.
В Екатеринбурге: ООО «РДЦ-Екатеринбург», ул. Прибалтийская, д. 24а.
Тел. +7 (343) 272-72-01/02/03/04/05/06/07/08.
В Новосибирске: ООО «РДЦ-Новосибирск», Комбинатский пер., д. 3. Тел. +7 (383) 289-91-42.
E-mail: eksmo-nsk@yandex.ru
В Киеве: ООО «РДЦ Эксмо-Украина», Московский пр-т, д. 6. Тел./факс: (044) 498-15-70/71.
В Донецке: ул. Артема, д. 160. Тел. +38 (062) 381-81-05.
В Харькове: ул. Гвардейцев Железнодорожников, д. 8. Тел. +38 (057) 724-11-56.
Во Львове: ул. Бузкова, д. 2. Тел. +38 (032) 245-01-71.
Интернет-магазин: www.knigka.ua. Тел. +38 (044) 228-78-24.
В Казахстане: ТОО «РДЦ-Алматы», ул. Домбровского, д. 3а. Тел./факс (727) 251-59-90/91.
RDC-Almaty@eksmo.kz

**Полный ассортимент продукции издательства «Эксмо»
можно приобрести в магазинах «Новый книжный» и «Читай-город».**
Телефон единой справочной: 8 (800) 444-8-444.
Звонок по России бесплатный.

Подписано в печать 25.06.2012. Формат 60×84¹/₈.
Гарнитура «Школьная». Печать офсетная. Усл. печ. л. 11,2.
Тираж экз. Заказ

ISBN 978-5-699-58095-8



9 785699 580958 >

ПОДГОТОВКА К ЕГЭ ВЫСШИЙ УРОВЕНЬ КАЧЕСТВА

ЕГЭ 2013



Успех на ЕГЭ гарантирован!

Для комплексной подготовки к ЕГЭ издательство «Эксмо» предлагает учебные пособия по всем предметам, которые выносятся на экзамен в 2013 году: русскому языку, литературе, математике, истории, обществознанию, биологии, географии, физике, химии, информатике и иностранным языкам.

Авторы пособий — ведущие специалисты и разработчики заданий ЕГЭ. Все книги прошли строгий контроль качества.

Комплект пособий поможет получить высший балл на ЕГЭ по информатике!



Аналогичные комплекты для подготовки к ЕГЭ выпускаются по всем предметам

ISBN 978-5-699-58095-8



9 785699 580958 >