

**ФЕДЕРАЛЬНЫЙ БАНК
ЭКЗАМЕНАЦИОННЫХ МАТЕРИАЛОВ**



ИНФОРМАТИКА

СБОРНИК ЭКЗАМЕНАЦИОННЫХ ЗАДАНИЙ



**ФЕДЕРАЛЬНЫЙ ИНСТИТУТ
ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ**

**ФЕДЕРАЛЬНЫЙ БАНК
ЭКЗАМЕНАЦИОННЫХ МАТЕРИАЛОВ**
(ОТКРЫТЫЙ СЕГМЕНТ)



2009

ИНФОРМАТИКА

СБОРНИК ЭКЗАМЕНАЦИОННЫХ ЗАДАНИЙ



МОСКВА ЭКСМО 2009



Авторы-составители:
П. А. Якушкин, С. С. Крылов

Е 33 **ЕГЭ 2009. Информатика. Федеральный банк экзамена-**
ционных материалов / Авт.-сост. П. А. Якушкин, С. С. Кры-
лов. — М. : Эксмо, 2009. — 160 с. + 1 CD-ROM. — (ЕГЭ. Феде-
ральный банк экзаменационных материалов).

ISBN 978-5-699-34431-4

В издании представлено более 500 экзаменационных заданий частей А, В и С. Задания подготовлены официальным разработчиком контрольных измерительных материалов — Федеральным институтом педагогических измерений — и сгруппированы по экзаменационным темам, соответствующим кодификатору ЕГЭ по информатике. По каждой теме предложены рекомендации и комментарии разработчиков заданий ЕГЭ, ко всем заданиям приведены ответы и критерии оценивания.

К изданию прилагается *диск с интерактивным тренинг-курсом*, позволяющим выпускнику и абитуриенту в обучающем режиме проверить свои знания, выбрав любую из экзаменационных тем или полный вариант контрольной работы. Диск также предоставляет возможность выполнять задания в экзаменационном режиме, фиксируя выбранные ответы в электронном экзаменационном бланке. И тренировочные, и экзаменационные варианты формируются из заданий федерального банка экзаменационных материалов в соответствии с кодификатором, спецификацией и планом экзаменационной работы.

Книга адресована *выпускникам* средней школы и *абитуриентам* для подготовки к единому государственному экзамену по информатике. Издание поможет *учителям* и *репетиторам* организовать эффективную подготовку учащихся к ЕГЭ.

+ CD

УДК 373.167.1 : 004
ББК 32.81я721

© ФГНУ «Федеральный институт педагогических измерений», 2009
© Материалы на компакт-диске.
ФГНУ «ФИПИ», 2009
© ООО «Издательство «Эксмо», 2009

ISBN 978-5-699-34431-4

ПРЕДИСЛОВИЕ

В 2008 году заканчивается эксперимент по введению единого государственного экзамена (ЕГЭ), а с 2009 года для всех выпускников и желающих поступить в вузы ЕГЭ станет обязательным. В ходе эксперимента на федеральном и региональном уровнях решались различные вопросы, связанные с процедурой проведения экзамена, поиском оптимальных экзаменационных моделей по общеобразовательным предметам, повышением качества контрольных измерительных материалов (КИМ).

В Российской Федерации созданием вариантов КИМ для ЕГЭ по 13 общеобразовательным предметам занимается только Федеральный институт педагогических измерений (ФИПИ). Ежегодная процедура разработки КИМ состоит из нескольких этапов, в том числе многократного проведения внешней экспертизы отдельных тестовых заданий и целых вариантов. Все тестовые задания ЕГЭ хранятся в Федеральном банке экзаменационных материалов (ФБЭМ). Для пополнения банка тестовыми заданиями ежегодно организуется конкурс «Контрольные измерительные материалы и тестовые задания для единого государственного экзамена». Пополнение банка необходимыми тестовыми материалами осуществляется также на основе целевого заказа. В настоящее время в ФБЭМ находится более 100 000 заданий.

По решению Коллегии Министерства образования и науки Российской Федерации каждый год ФИПИ после проведения экзамена открывает несколько вариантов по всем общеобразовательным предметам. Задания из этих вариантов, по поручению Федеральной службы по надзору в сфере образования и науки, размещаются на сайте ФИПИ (www.fipi.ru) в открытом сегменте ФБЭМ.

В данном пособии открытый сегмент ФБЭМ представлен в соответствии с содержательными линиями школьного курса и отражает особенности структуры КИМ по всем предметам. Каждый блок заданий сопровождается методическим комментарием, необходимым для эффективной подготовки к ЕГЭ.

Издания серии «Федеральный банк экзаменационных материалов» будут полезны как выпускникам и абитуриентам, желающим получить полное представление о содержании экзамена и всех типах заданий, входящих в состав ЕГЭ, так и учителям, которые с помощью опубликованных контрольных измерительных материалов по всем экзаменационным темам смогут организовать различные формы текущего и обобщающего контроля.

Директор Федерального института
педагогических измерений
А.Г. Ершов

ВВЕДЕНИЕ

Целью единого государственного экзамена по информатике является объективная оценка общеобразовательной подготовки по этому предмету выпускников XI (XII) классов общеобразовательных учреждений и абитуриентов с целью отбора для зачисления в учреждения высшего профессионального образования. В 2006 г. впервые ЕГЭ по информатике в одном регионе (г. Санкт-Петербург) проходил не только в рамках «второй, абитуриентской волны», но и в июне, в рамках единого экзамена. В июне 2007 г. экзамен для выпускников прошел уже в 13 регионах страны, а в 2009 г. он уже должен быть проведен во всех субъектах Российской Федерации.

Содержание экзаменационной работы определяется на основе утвержденного Министерством образования и науки Российской Федерации обязательного минимума содержания среднего (полного) общего образования по информатике (приказ от 30.06.99 № 56).

Экзаменационная работа состоит из трех частей. Часть 1 (А) содержит 20 заданий из всех тематических блоков, кроме заданий по технологии телекоммуникаций и технологии программирования. Эти задания предполагали выбор одного ответа из четырех предложенных.

Часть 2 (В) включает задания по темам: «Информация и ее кодирование», «Основы логики», «Алгоритмизация и программирование», «Телекоммуникационные технологии» — всего 8 заданий с кратким ответом.

Задания части 3 (С) направлены на проверку сформированности важнейших умений записи и анализа алгоритмов, предусмотренных требованиями к обязательному уровню подготовки по информатике учащихся средних общеобразовательных учреждений. В этой части также проверялись умения на повышенном и высоком уровне сложности по теме «Технология программирования». Решения заданий третьей части работы записываются в развернутой форме и проверяются экспертами региональных предметных комиссий. За выполнение каждого задания дается определенное количество баллов, в зависимости от полноты и качества выполнения, поэтому задания части 3 при общем количестве заданий 4 (12,5% общего количества заданий) при макси-

мальном выполнении дают 12 первичных баллов, что составляет 30% общего количества первичных баллов. С другой стороны, эти задания являются сложными и самыми трудоемкими: рекомендованное время их выполнения в 2 раза превосходит время, отводимое на выполнение первых двух частей работы.

Содержание экзамена включает основные темы курса информатики и информационных технологий, объединенных в следующие тематические блоки: «Информация и ее кодирование», «Алгоритмизация и программирование», «Основы логики», «Моделирование и компьютерный эксперимент», «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». Распределение заданий по разделам курса информатики представлено в таблице.

Распределение заданий по разделам курса информатики

№ п/п	Название раздела	Вся работа		Часть 1		Часть 2		Часть 3	
		К-во заданий	%	К-во заданий	%	К-во заданий	%	К-во заданий	%
1	Информация и ее кодирование	8	25	6	19	2	6	0	0
2	Алгоритмизация и программирование	9	28	5	16	2	6	2	6
3	Основы логики	5	16	3	9	2	6	0	0
4	Моделирование	1	3	1	3	0	0	0	0
5	Программные средства информационных и коммуникационных технологий	1	3	1	3	0	0	0	0
6	Технология обработки графической и звуковой информации	1	3	1	3	0	0	0	0
7	Технология обработки информации в электронных таблицах	2	6	2	6	0	0	0	0

№ п/п	Название раздела	Вся работа		Часть 1		Часть 2		Часть 3	
		К-во зада- ний	%	К-во зада- ний	%	К-во зада- ний	%	К-во зада- ний	%
8	Технология хранения, поиска и сортировки информации в базах данных	1	3	1	3	0	0	0	0
9	Телекоммуникационные технологии	2	6	0	0	2	6	0	0
10	Технология программирования	2	6	0	0	0	0	2	6
	Итого:	32	99	20	62	8	24	4	12

Экзамен проверяет знания и умения выпускников на различных уровнях. Базовый уровень представляет собой задания на проверку знаний и умений инвариантной составляющей курса информатики, преподающегося в классах и учебных заведениях всех профилей. Таких заданий в работе ровно половина. Задания повышенного уровня связаны с содержанием профильных курсов информатики, требующих более углубленного изучения. Задания высокого уровня призваны выделить учащихся, хорошо овладевших содержанием учебного предмета, ориентированных на получение высшего профессионального образования в областях, связанных с информатикой и компьютерной техникой. Важным вопросом, учитываемым при разработке контрольных измерительных материалов для экзамена, является способ и уровень проверки знаний: какова доля заданий на простое воспроизведение материала, в какой ситуации проверяется умение применять полученные знания. В КИМ по информатике практически отсутствуют задания, требующие простого воспроизведения знания терминов, понятий, величин, правил. В любом случае экзаменуемому требуется решить какую-либо задачу: либо прямо использовать известное правило, алгоритм, умение, либо выбрать из общего количества изученных понятий и алгоритмов наиболее подходящее и применить его в известной или новой ситуации. Таким образом, речь идет об уровне, на котором выполняется то или иное задание.

На уровне *воспроизведения знаний* через несложные задания в одно-два действия проверяется фундаментальный теоретический материал, такой, как:

- единицы измерения информации;
- принципы кодирования;
- системы счисления;
- понятие алгоритма, его свойств, способов записи;
- основные алгоритмические конструкции;
- основные элементы программирования;
- основные элементы математической логики;
- основные типы информационных моделей;
- программное обеспечение;
- основные понятия, используемые в информационных и коммуникационных технологиях.

Материал на проверку сформированности *умений применять свои знания в стандартной ситуации*, входящий во все три части экзаменационной работы, включает следующие умения:

- подсчитывать информационный объем сообщения;
- осуществлять перевод из одной системы счисления в другую;
- осуществлять арифметические действия в двоичной, восьмеричной и шестнадцатеричной системах счисления;
- использовать стандартные алгоритмические конструкции при программировании;
- формально исполнять алгоритмы, записанные на естественных и алгоритмических языках, в том числе в виде блок-схем и на языках программирования;
- создавать и преобразовывать логические выражения;
- формировать для логической функции таблицу истинности и логическую схему;
- оценивать результат работы известного программного обеспечения;
- формулировать запросы к базам данных и поисковым системам.

Материал на проверку сформированности *умений применять свои знания в новой ситуации*, входящий во вторую и третью часть работы, включает следующие комплексные умения:

- решать сложные логические высказывания;
- анализировать текст программы с точки зрения соответствия записанного алгоритма поставленной задаче и изменять его в соответствии с заданием;
- реализовывать сложный алгоритм с использованием современных систем программирования.

Как уже отмечалось, экзаменационная работа состоит из трех частей. Первая представляет собой 20 заданий с выбором одного верного ответа из четырех предложенных. Каждый верный ответ оценивается одним баллом. Вторая часть работы состоит из восьми заданий, ответ на которые представляет собой какую-то строку, число или последовательность чисел. Учащиеся формулируют ответ и записывают его в поле для ответа на специальном бланке. Бланки с ответами на задания первой и второй части работы сканируются и затем централизованно автоматически обрабатываются.

Естественно, что в отличие от результатов обработки первой части работы, где ошибки распознавания исключены, при сканировании и распознавании ответов на задания второй части возможны ошибки обработки. Самыми распространенными ошибками распознавания являются смешение букв и цифр (цифра 2 и буква Г, цифра 4 и буква Ч), неправильное распознавание букв и т.д. Задания второй части специально составлены таким образом, чтобы исключить смешение символов, поэтому результаты компьютерной обработки вариантов (т.н. «веер ответов») просматриваются экспертами с точки зрения признания верными ответов, отбракованных компьютером в результате ошибки распознавания. Тем не менее в экзаменационных работах достаточно много ошибок, связанных с неправильным заполнением бланка ответов на вопросы второй части работы. При подготовке учащихся к экзамену следует обратить их внимание на то, что все задания второй части очень точно формулируют требования к формату записи ответа: в каком порядке записывать перечисление чисел, какие пробелы и знаки препинания ставить и т.п. Также на уроках информатики можно объяснить учащимся всю сложность задачи распознавания письменного текста и тем самым проиллюстрировать необходимость записывать ответ с помощью букв и цифр стандартной формы, максимально соответствующих образцу, приведенному на бланке для записи ответов.

Ответы на задания третьей части работы записываются учащимися на специальных бланках, которые потом сканируются и распечатываются в графическом формате (без распознавания) для передачи экспертам на проверку. Все отсканированные ответы на задания третьей части затем хранятся в цифровом виде в центре проведения ЕГЭ, так что качество работы экспертов может быть оценено в любое время после завершения экзамена. (В этом смысле процедура проведения ЕГЭ гораздо прозрачнее

обычных устных экзаменов.) Эксперты, проверяющие ответы на задания третьей части, назначаются региональной предметной комиссией из числа специально подготовленных учителей школ и преподавателей высших учебных заведений. Каждый эксперт проверяет определенное количество работ, строго руководствуясь единой для всей страны инструкцией и едиными критериями оценивания. Критерии оценивания вырабатываются Федеральной предметной комиссией и рассылаются в регионы вместе с вариантами работ. Каждая работа проверяется двумя экспертами. В случае значительного (более чем в 1 балл) расхождения оценок экспертов, региональная комиссия передает работы на перепроверку третьему эксперту, чье мнение считается окончательным. Иначе ставится более высокая (из двух) оценка.

Готовя учащихся к экзамену, надо еще раз обратить их внимание на то, что ответы на задания третьей части работы должны быть записаны четко, понятным почерком, в строгом соответствии с требованиями, сформулированными в задании. Иначе резко возрастает трудоемкость оценивания работы.

ЭКЗАМЕНАЦИОННЫЕ ЗАДАНИЯ

РАЗДЕЛ 1. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

Информация и ее кодирование¹

1. /1.1.8/ Считая, что каждый символ кодируется 16-ю битами, оцените информационный объем следующей пушкинской фразы в кодировке Unicode:
Привычка свыше нам дана: Замена счастию она.
1) 44 бита 2) 704 бита 3) 44 байта 4) 704 байта
2. /1.1.8/ Каждый символ в Unicode закодирован двухбайтным словом. Оцените информационный объем следующего предложения в этой кодировке:
Без труда не вытащишь рыбку из пруда.
1) 37 бит 2) 592 бита 3) 37 байт 4) 592 байта
3. /1.1.8/ Каждый символ в Unicode закодирован двухбайтным словом. Оцените информационный объем следующего предложения в этой кодировке:
Без охоты не спору у работы.
1) 28 байт 2) 28 бит 3) 448 байт 4) 448 бит
4. /1.1.3/ Световое табло состоит из лампочек, каждая из которых может находиться в двух состояниях («включено» или «выключено»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 200 разных сигналов?
1) 8 2) 9 3) 100 4) 200

¹ После порядкового номера в скобках указан номер данной проверяемой позиции по кодификатору элементов содержания по информатике для составления контрольных измерительных материалов 2007 года.

5. /1.1.3/ Световое табло состоит из лампочек, каждая из которых может находиться в двух состояниях («включено» или «выключено»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 100 различных сигналов?
- 1) 6 2) 7 3) 50 4) 100
6. /1.1.3/ Световое табло состоит из лампочек, каждая из которых может находиться в двух состояниях («включено» или «выключено»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 50 различных сигналов?
- 1) 5 2) 6 3) 25 4) 50
7. /1.1.3/ Какое наименьшее число символов должно быть в алфавите, чтобы при помощи всевозможных трехбуквенных слов, состоящих из символов данного алфавита, можно было передать не менее 9 различных сообщений?
- 1) 5 2) 2 3) 3 4) 4
8. /1.1.3/ В соревновании участвуют 215 атлетов. Какое минимальное количество бит необходимо, чтобы кодировать номер каждого атлета?
- 1) 8 2) 14 3) 26 4) 27
9. /1.1.4/ Для передачи секретного сообщения используется код, состоящий из десятичных цифр. При этом все цифры кодируются одним и тем же (минимально возможным) количеством бит. Определите информационный объем сообщения длиной в 150 символов.
- 1) 600 бит 2) 750 бит 3) 1200 бит 4) 60 байт
10. /1.1.4/ Для передачи секретного сообщения используется код, состоящий из заглавных латинских букв и цифр (всего используется 36 различных символов). При этом все символы кодируются одним и тем же (минимально возможным) количеством бит. Определите информационный объем сообщения длиной в 150 символов.
- 1) 900 бит 2) 160 байт 3) 800 бит 4) 120 байт

11. /1.1.4/ Метеорологическая станция ведет наблюдение за влажностью воздуха. Результатом одного измерения является целое число от 0 до 100%, которое записывается при помощи минимально возможного количества бит. Станция сделала 80 измерений. Определите информационный объем результатов наблюдений.
- 1) 80 бит 2) 70 байт 3) 80 байт 4) 560 байт
12. /1.1.4/ Метеорологическая станция ведет наблюдение за направлением ветра. Результатом одного измерения является одно из 8 возможных направлений, которое записывается при помощи минимально возможного количества бит. Станция сделала 160 измерений. Каков информационный объем результатов наблюдений?
- 1) 160 байт 2) 80 байт 3) 60 байт 4) 160 бит
13. /1.1.4/ Метеорологическая станция ведет наблюдение за атмосферным давлением. Результатом одного измерения является целое число, принимающее значение от 720 до 780 мм ртутного столба, которое записывается при помощи минимально возможного количества бит. Станция сделала 80 измерений. Каков информационный объем результатов наблюдений?
- 1) 60 байт 2) 80 бит 3) 80 байт 4) 480 байт
14. /1.1.4/ Азбука Морзе позволяет кодировать символы для радиосвязи, задавая комбинацию точек и тире. Сколько различных символов (цифр, букв, знаков пунктуации и т.д.) можно закодировать, используя код Морзе длиной не менее пяти и не более шести сигналов (точек и тире)?
- 1) 80 2) 120 3) 112 4) 96
15. /1.1.4/ Сколько бит информации содержит сообщение объемом 4 мегабайта?
- 1) 2^{15} 2) 2^{25} 3) 2^{30} 4) $32 \cdot 10^6$
16. /1.1.4/ Сколько мегабайт информации содержит сообщение объемом 2^{23} бит?
- 1) 1 2) 8 3) 3 4) 32

17. /1.1.7/ Сколько единиц в двоичной записи числа 195?
1) 5 2) 2 3) 3 4) 4
18. /1.1.7/ Сколько единиц в двоичной записи числа 197?
1) 5 2) 2 3) 3 4) 4
19. /1.1.7/ Количество значащих нулей в двоичной записи десятичного числа 129 равно:
1) 5 2) 6 3) 7 4) 4
20. /1.1.7/ Количество значащих нулей в двоичной записи десятичного числа 129 равно:
1) 5 2) 6 3) 7 4) 4
21. /1.1.7/ Количество значащих нулей в двоичной записи десятичного числа 132 равно:
1) 1 2) 2 3) 3 4) 0
22. /1.1.7/ Как представлено число 75_{10} в двоичной системе счисления?
1) 1001011_2 2) 100101_2 3) 1101001_2 4) 111101_2
23. /1.1.7/ Количество цифр в двоичной записи десятичного числа, которое можно представить в виде $2 + 8 + 16 + 64 + 128 + 256 + 512$, равно
1) 7 2) 8 3) 9 4) 10
24. /1.1.7/ Вычислите сумму чисел x и y , при $x = B4_{16}$, $y = 46_8$.
Результат представьте в двоичной системе счисления.
1) 11011010_2 2) 10000010_2 3) 1110010_2 4) 10111010_2
25. /1.1.7/ Значение выражения $10_{16} + 10_8 \cdot 10_2$ в двоичной системе счисления равно:
1) 1010 2) 11010 3) 100000 4) 110000
26. /1.1.7/ Вычислите сумму чисел x и y , при $x = A7_{16}$, $y = 56_8$.
Результат представьте в двоичной системе счисления.
1) 11010101_2 2) 11001001_2 3) 10001111_2 4) 10000101_2

27. /1.1.7/ Вычислите сумму чисел x и y , при $x = 1D_{16}$, $y = 72_8$.
Результат представьте в двоичной системе счисления.
- 1) 10001111_2 2) 1100101_2 3) 101011_2 4) 1010111_2
28. /1.1.7/ Укажите наименьшее основание системы счисления, в которой запись числа 19 трехзначна.
29. /1.1.7/ Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 22 оканчивается на 4.
30. /1.1.7/ Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 24 оканчивается на 3.
31. /1.1.7/ Укажите через запятую в порядке возрастания все числа, не превосходящие 25, запись которых в двоичной системе счисления оканчивается на 101. Ответ запишите в десятичной системе счисления.
32. /1.1.7/ В системе счисления с некоторым основанием число 17 записывается в виде 101. Укажите это основание.
33. /1.1.5/ Для кодирования букв А, Б, В, Г решили использовать двухразрядные последовательные двоичные числа (от 00 до 11 соответственно). Если таким способом закодировать последовательность символов ГБАВ и записать результат шестнадцатиричным кодом, то получится:
- 1) D2 2) 132 3) 3102 4) DBAC
34. /1.1.5/ Для кодирования букв А, Б, В, Г решили использовать двухразрядные последовательные двоичные числа (от 00 до 11 соответственно). Если таким способом закодировать последовательность символов ВГБА и записать результат шестнадцатиричным кодом, то получится:
- 1) CDBA; 2) 114; 3) 2310; 4) B4.
35. /1.1.5/ Для кодирования букв А, Б, В, Г решили использовать двухразрядные последовательные двоичные числа (от 00 до 11 соответственно). Если таким способом закодировать последовательность символов ГБАВ и записать результат шестнадцатиричным кодом, то получится:
- 1) 138; 2) DBCA; 3) D8; 4) 3120.

36. /1.1.5/ Для 5 букв русского алфавита заданы их двоичные коды (для некоторых букв — из двух бит, для некоторых — из трех). Эти коды представлены в таблице:

В	К	А	Р	Д
000	11	01	001	10

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

- 1) 110100000100110011
 - 2) 111010000010010011
 - 3) 110100001001100111
 - 4) 110110000100110010
37. /1.1.6/ Сколько секунд потребуется модему, передающему информацию со скоростью 32 000 бит/с, чтобы передать 16-цветное растровое изображение размером 800×600 пикселей, при условии, что в одном байте закодировано максимально возможное целое число пикселей?
38. /1.1.6/ Скорость передачи данных через ADSL-соединение равна 256 000 бит/с. Передача файла через данное соединение заняла 3 мин. Определите размер файла в килобайтах.
39. /1.1.6/ Скорость передачи данных через ADSL-соединение равна 512 000 бит/с. Передача файла через данное соединение заняла 1 мин. Определите размер файла в килобайтах.
40. /1.1.6/ Известно, что длительность непрерывного подключения к сети Интернет с помощью модема для некоторых АТС не превышает 10 мин. Определите максимальный размер файла (в килобайтах), который может быть передан за время такого подключения, если модем передает информацию в среднем со скоростью 32 Кбит/с?
41. /1.2.1/ Дешифровщику необходимо восстановить поврежденный фрагмент сообщения, состоящий из 4-х символов.

Имеется достоверная информация, что использовано не более пяти букв (X, Z, Y, V, E), причем на четвертом месте стоит один из символов X, Z, Y. На третьем — гласная буква (Y, E), если

четвертая буква согласная, или любая согласная, если четвертая гласная. На втором месте – одна из букв *Z, Y, X, V*, не стоящая в слове на первом или четвертом месте. На первом месте — любая согласная буква, не стоящая на третьем месте.

Появилась дополнительная информация, что возможен один из четырех вариантов. Какой?

- 1) *VZYZ* 2) *YVEZ* 3) *VYYZ* 4) *ZVZY*

42. /1.2.1/ Дешифровщику необходимо восстановить поврежденный фрагмент сообщения, состоящий из 4-х символов.

Имеется достоверная информация, что использовано не более пяти букв (*A, B, C, D, E*), причем на третьем месте стоит один из символов *C, D, E*. На четвертом месте – одна из букв *A, C, D*, не стоящая на третьем месте. На первом месте — одна из букв *B, C, D, E*, не стоящая в слове на втором или четвертом месте. На втором – любая согласная буква, если третья буква гласная, или любая гласная, если третья согласная.

Появилась дополнительная информация, что возможен один из четырех вариантов. Какой?

- 1) *BADD* 2) *BCDC* 3) *DECD* 4) *DEDC*

Алгоритмизация и программирование

43. /1.2.1/ Цепочка из трех бусин формируется по следующему правилу. На третьем месте в цепочке стоит одна из бусин *A, B, Г*. На втором — одна из бусин *A, Б, В*. На первом месте — одна из бусин *Б, В, Г*, не стоящая в цепочке на втором или третьем месте.

Какая из следующих цепочек создана по этому правилу?

- 1) *БГВ* 2) *ГБА* 3) *ВАБ* 4) *ГВГ*

44. /1.2.1/ Для составления цепочек разрешается использовать бусины 5 типов, обозначаемых буквами *A, Б, В, Е, И*. Каждая цепочка должна состоять из трех бусин, при этом должны соблюдаться следующие правила:

- 1) на первом месте стоит одна из букв: *A, Е, И*;
- 2) после гласной буквы в цепочке не может снова идти гласная, а после согласной — согласная;
- 3) последней буквой не может быть *A*.

Какая из цепочек построена по этим правилам?

- 1) *АИБ* 2) *ЕВА* 3) *БИВ* 4) *ИБИ*

45. /1.2.1/ Для составления цепочек используются бусины, помеченные буквами А, Б, В, Г, Д, Е, по следующему правилу.

На первом месте в цепочке стоит одна из бусин А, Б, В, Г. На втором — любая согласная буква, если первая буква согласная, или любая гласная, если первая гласная. На третьем месте — одна из бусин Б, В, Д, Е, не стоящая в цепочке на первом или втором месте. На четвертом месте — любая согласная буква, не стоящая на втором или третьем месте.

Какая из перечисленных цепочек создана по этому правилу?

- 1) БАДБ 2) ББДГ 3) БАДБ 4) БДБГ

46. /1.2.1/ Для составления цепочек используются бусины, помеченные буквами: М, N, O, P, S. В середине цепочки стоит одна из бусин М, O, S. На третьем — любая гласная, если первая буква согласная, и любая согласная, если первая гласная. На первом месте — одна из бусин O, P, S, не стоящая в цепочке в середине.

Какая из перечисленных цепочек создана по этому правилу?

- 1) SMP 2) MSO 3) SNO 4) OSN

47. /1.2.1/ Строки (цепочки символов латинских букв) создаются по следующему правилу.

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

Вот первые 4 строки, созданные по этому правилу:

- (1) А
(2) ВАА
(3) СВААВАА
(4) DCBAABAACBAABAА

Латинский алфавит (для справки):

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Запишите восемь символов подряд, стоящих в седьмой строке со 119-го по 126-е место (считая слева направо).

48. /1.2.1/ Цепочки символов (строки) создаются по следующему правилу.

Первая строка состоит из одного символа — цифры «1».

Каждая из последующих цепочек создается следующим действием: в очередную строку дважды записывается предыдущая цепочка цифр (одна за другой, подряд), а в конец приписывается еще одно число — номер строки по порядку (на i -м шаге дописывается число « i »).

Вот первые 4 строки, созданные по этому правилу:

- 1) 1
- 2) 112
- 3) 1121123
- 4) 112112311211234

Сколько раз в общей сложности встречаются в восьмой строке четные цифры (2, 4, 6, 8)?

49. /1.2.1/ Цепочки символов (строки) создаются по следующему правилу.

Первая строка состоит из одного символа — цифры «1».

Каждая из последующих цепочек создается следующим действием: в очередную строку дважды записывается предыдущая цепочка цифр (одна за другой, подряд), а в конец приписывается еще одно число — номер строки по порядку (на i -м шаге дописывается число « i »).

Вот первые 4 строки, созданные по этому правилу:

- 1) 1
- 2) 112
- 3) 1121123
- 4) 112112311211234

...

Сколько раз в общей сложности встречаются в восьмой строке нечетные цифры (1, 3, 5, 7, 9)?

50. /1.2.1/ Цепочки символов (строки) создаются по следующему правилу.

Первая строка состоит из одного символа — цифры «1».

Каждая из последующих цепочек создается следующим действием: в очередную строку дважды записывается предыдущая цепочка цифр (одна за другой, подряд), а в конец приписывается еще одно число — номер строки по порядку (на i -м шаге дописывается число « i »).

Вот первые 4 строки, созданные по этому правилу:

- 1) 1
- 2) 112
- 3) 1121123
- 4) 112112311211234

...

Сколько раз в общей сложности встречаются в девятой строке четные цифры (2, 4, 6, 8)?

51. /1.2.1/ Цепочки символов (строки) создаются по следующему правилу.

Первая строка состоит из одного символа — цифры «1».

Каждая из последующих цепочек создается такими действиями: в очередную строку дважды записывается цепочка цифр из предыдущей строки (одна за другой, подряд), а в конец приписывается еще одно число — номер строки по порядку (на i -м шаге дописывается число « i »).

Вот первые 4 строки, созданные по этому правилу:

- 1) 1
- 2) 112
- 3) 1121123
- 4) 112112311211234

Какая цифра стоит в седьмой строке на 120-м месте (считая слева направо)?

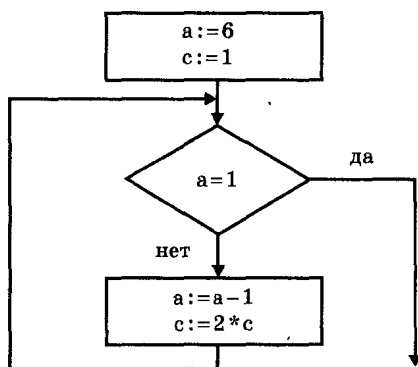
52. /1.2.1/ Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 1, а во второй — 2 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или увеличивает в 3 раза число камней в какой-то кучке, или добавляет 3 камня в какую-то кучку. Выигрывает игрок, после хода которого в одной из кучек становится не менее 24 камней. Кто выигрывает при безошибочной игре — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

53. /1.2.1/ Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 3, а во второй — 2 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок

или увеличивает в 3 раза число камней в какой-то кучке, или добавляет 3 камня в какую-то кучку. Выигрывает игрок, после хода которого в одной из кучек становится не менее 24 камней. Кто выигрывает при безошибочной игре — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

54. /1.2.1/ Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 3, а во второй — 2 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или увеличивает в 3 раза число камней в какой-то кучке, или добавляет 1 камень в какую-то кучку. Выигрывает игрок, после хода которого общее число камней в двух кучках становится не менее 16 камней. Кто выигрывает при правильной игре — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

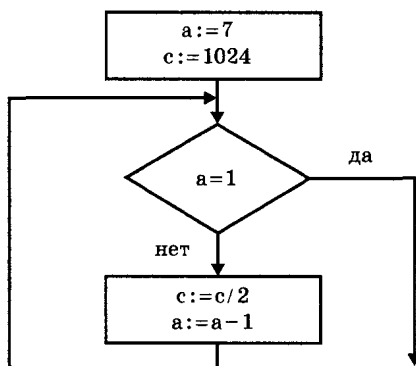
55. /1.2.2/ Определите значение переменной s после выполнения фрагмента алгоритма:



Примечание: знаком $:=$ обозначена операция присваивания, знаком $*$ обозначена операция умножения.

- 1) 16 2) 2 3) 32 4) 64

56. /1.2.2/ Определите значение переменной s после выполнения фрагмента алгоритма:



Примечание: знаком $:=$ обозначена операция присваивания.

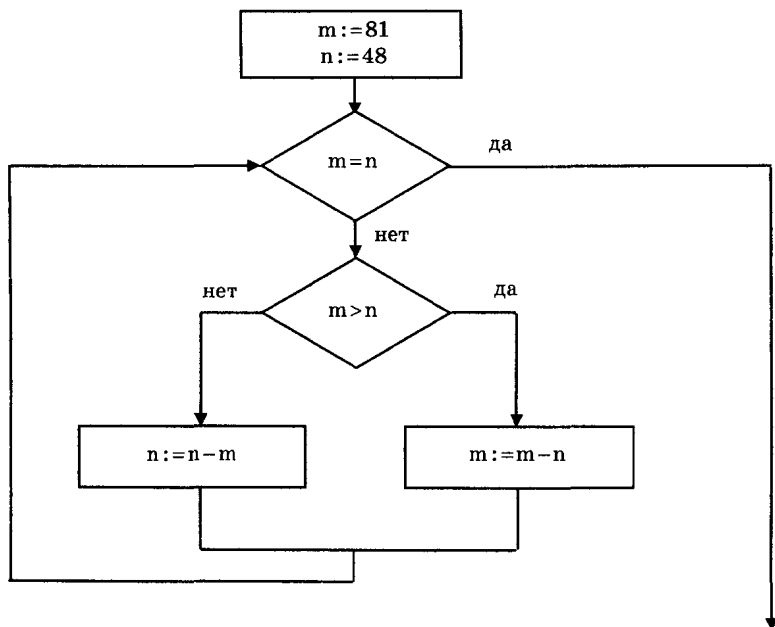
1) 32

2) 16

3) 8

4) 4

57. /1.2.2/ Определите значение переменной m после выполнения фрагмента алгоритма:



Примечание: знаком $:=$ обозначена операция присваивания.

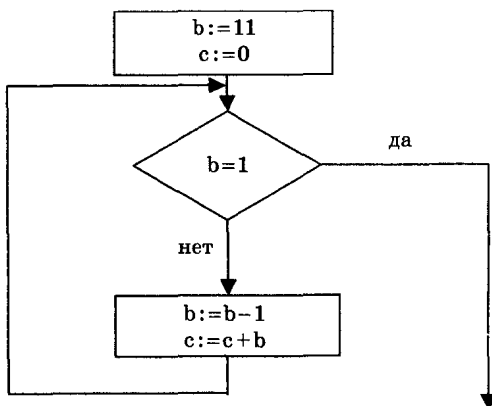
1) 1

2) 2

3) 3

4) 33

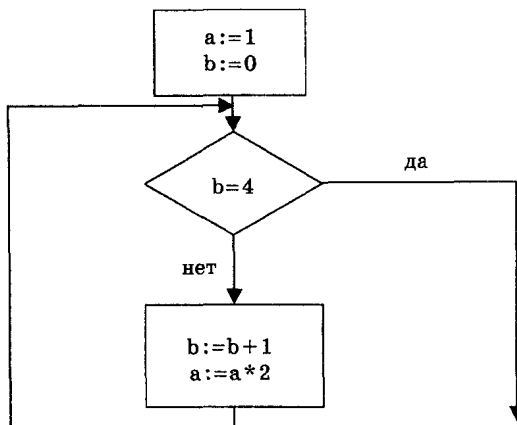
58. /1.2.2/ Определите значение переменной c после выполнения фрагмента алгоритма:



Примечание: знаком $:=$ обозначена операция присваивания.

- 1) 66 2) 100 3) 55 4) 54

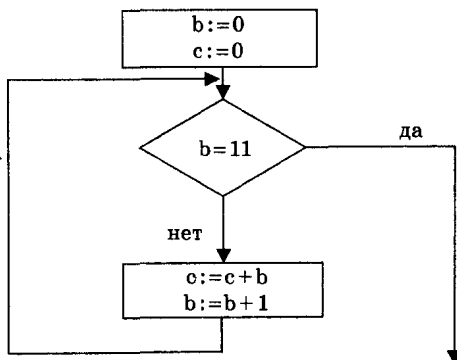
59. /1.2.2/ Определите значение переменной a после выполнения фрагмента алгоритма:



Примечание: знаком $*$ обозначено умножение, знаком $:=$ обозначена операция присваивания.

- 1) 8 2) 16 3) 32 4) 12

60. /1.2.2/ Определите значение переменной c после выполнения фрагмента алгоритма:

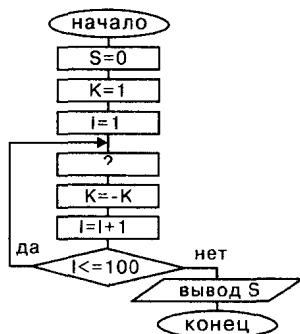


Примечание: знаком := обозначена операция присваивания.

- 1) 1 2) 45 3) 55 4) 66

61. /1.2.2/ Какая команда должна быть в блок-схеме подсчета суммы:

$S = 1 - 2 + 3 - 4 + \dots - 100$ на месте вопросительного знака?



- 1) $S = S + I$ 2) $S = S + K$ 3) $S = I * K$ 4) $S = S + I * K$

62. /1.2.2/ В приведенном ниже фрагменте алгоритма, записанном на алгоритмическом языке, переменные a, b, c имеют тип «строка», а переменные i, k — тип «целое». Используются следующие функции:

Длина (a) — возвращает количество символов в строке a . (Тип «целое»)

Извлечь (a, i) — возвращает i -й (слева) символ в строке a . (Тип «строка».)

Склеить (a, b) — возвращает строку, в которой записаны сначала все символы строки a , а затем все символы строки b . (Тип «строка».)

Значения строк записываются в одинарных кавычках. (Например, $a := \text{'дом'}$.)

Фрагмент алгоритма:

```

 $i := \text{Длина}(a)$ 
 $k := 1$ 
 $b := \text{'П'}$ 
пока  $i > 0$ 
нц
 $c := \text{Извлечь}(a, i)$ 
 $b := \text{Склеить}(b, c)$ 
 $i := i - k$ 
кц

```

Какое значение будет у переменной b после выполнения вышеприведенного фрагмента алгоритма, если значение переменной a было 'РОЗА'?

- 1) 'ПАЗ' 2) 'ПАЗОР' 3) 'ПОЗА' 4) 'ПРОЗА'

63. /1.2.2/ В приведенном ниже фрагменте алгоритма, записанном на алгоритмическом языке, переменные a, b, c имеют тип «строка», а переменные i, k — тип «целое». Используются следующие функции:

Длина (a) — возвращает количество символов в строке a . (Тип «целое».)

Извлечь (a, i) — возвращает i -й (слева) символ в строке a . (Тип «строка».)

Склеить (a, b) — возвращает строку, в которой записаны сначала все символы строки a , а затем все символы строки b . (Тип «строка».)

Значения строк записываются в одинарных кавычках. (Например, $a := \text{'дом'}$.)

Фрагмент алгоритма:

```

 $i := \text{Длина}(a)$ 
 $k := 2$ 
 $b := \text{'А'}$ 
пока  $i > 0$ 
нц
 $c := \text{Извлечь}(a, i)$ 
 $b := \text{Склеить}(b, c)$ 
 $i := i - k$ 
кц
 $b := \text{Склеить}(b, \text{'Т'})$ 

```

Какое значение будет у переменной **b** после выполнения выше-приведенного фрагмента алгоритма, если значение переменной **a** было 'ПОЕЗД'.

- 1) 'АДЕПТ' 2) 'АДЗЕОП' 3) 'АДТЕПТ' 4) 'АДЗОТ'

64. /1.2.2/ В приведенном ниже фрагменте алгоритма, записанном на алгоритмическом языке, переменные **a**, **b**, **c** имеют тип «строка», а переменные **i**, **n**, **k** — тип «целое». Используются следующие функции:

Длина (**a**) — возвращает количество символов в строке **a**. (Тип «целое»)

Извлечь (**a**, **i**) — возвращает **i**-тый (слева) символ в строке **a**. (Тип «строка»)

Склеить (**a**, **b**) — возвращает строку, в которой записаны сначала все символы строки **a**, а затем все символы строки **b**. (Тип «строка»)

Значения строк записываются в одинарных кавычках (Например, **a** := 'дом').

Фрагмент алгоритма:

```

n := Длина (a)
k := 2
i := 1
b := 'Р'
пока i < n
  нц
    c := Извлечь (a, i)
    b := Склеить (b, c)
    i := i + k
  кц
b := Склеить (b, 'ДА')

```

Какое значение будет у переменной **b** после выполнения выше-приведенного фрагмента алгоритма, если значение переменной **a** было 'МОТОР'?

- 1) 'РМТДА' 2) 'РОТОДА' 3) 'РОТОМДА' 4) 'РОТОНДА'

65. /1.2.2/ У исполнителя *Калькулятор* две команды, которым присвоены номера:

- 1) прибавь 2;
2) умножь на 3.

Выполняя первую из них, *Калькулятор* прибавляет к числу на экране 2, а выполняя вторую, утраивает его. Запишите поря-

док команд в программе получения из 0 числа 28, содержащей не более 6 команд, указывая лишь номера команд. (Например, программа 21211 — это программа:

умножь на 3;
прибавь 2;
умножь на 3;
прибавь 2;
прибавь 2,
которая преобразует число 1 в 19.)

66. /1.2.2/ У исполнителя *Квадратор* две команды, которым присвоены номера:

- 1) возведи в квадрат;
- 2) прибавь 1.

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

возведи в квадрат;
прибавь 1;
возведи в квадрат;
прибавь 1;
прибавь 1,
которая преобразует число 1 в 6.)

67. /1.2.2/ У исполнителя *Калькулятор* две команды, которым присвоены номера:

- 1) прибавь 3;
- 2) умножь на 2.

Выполняя первую из них, *Калькулятор* прибавляет к числу на экране 3, а выполняя вторую, удваивает его. Запишите порядок команд в программе получения из 1 числа 47, содержащей не более 6 команд, указывая лишь номера команд. (Например, программа 21211 — это программа:

умножь на 2;
прибавь 3;
умножь на 2;
прибавь 3;
прибавь 3,
которая преобразует число 1 в 16.)

68. /1.2.2/ У исполнителя *Утроитель* две команды, которым присвоены номера:

1) **вычти 2;**

2) **умножь на три.**

Первая из них уменьшает число на экране на 2, вторая — утраивает его. Запишите порядок команд в программе получения из 11 числа 13, содержащей не более 5 команд, указывая лишь номера команд. (Например, **21211** — это программа:

умножь на три

вычти 2

умножь на три

вычти 2

вычти 2,

которая преобразует число 2 в 8.)

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

69. /1.2.2/ 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, вызывающая передвижение черепашки на n шагов в направлении движения.

Направо m , где m — целое число, вызывающая изменение направления движения на m градусов по часовой стрелке.

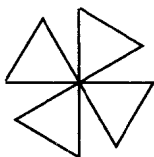
Запись **Повтори 5 [Команда1 Команда2]** означает, что последовательность команд в скобках повторится 5 раз. Исполнитель интерпретирует эту запись как одну команду.

Черепашке был дан для исполнения следующий алгоритм:

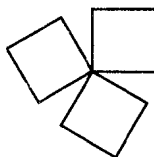
Повтори 5 [Повтори 4 [Вперед 40 Направо 90] Направо 120]

Какая фигура появится на экране?

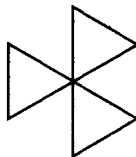
1)



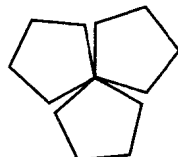
2)



3)



4)



70. /1.2.2/ 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 , вызывающая передвижение *Черепашки* на n шагов в направлении движения.

Направо m , вызывающая изменение направления движения на m градусов по часовой стрелке. $0 \leq m \leq 180$.

(Вместо n и m должны стоять целые числа.)

Запись:

Повтори 5 [Команда1 Команда2]

означает, что последовательность команд в квадратных скобках повторится 5 раз.

Какое число нужно записать вместо n в следующем алгоритме:

Повтори 7 [Вперед 40 Направо n],

чтобы на экране появился правильный шестиугольник?

71. /1.2.2/ Исполнитель Робот действует на клетчатой доске, между соседними клетками которой могут стоять стены. *Робот* передвигается по клеткам доски и может выполнять команды 1 (вверх), 2 (вниз), 3 (вправо), 4 (влево), переходя на соседнюю клетку в направлении, указанном в скобках. Если в этом направлении между клетками стоит стена, то *Робот* разрушается. *Робот* успешно выполнил программу 33233241.

Какую последовательность из четырех команд должен выполнить *Робот*, чтобы вернуться в ту клетку, где он был перед началом выполнения программы, и не разрушиться вне зависимости от того, какие стены стоят на поле?

72. /1.2.2/ Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

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

При выполнении этой команды РОБОТ перемещается на соответствующую клетку.

Команды проверки истинности условия на наличие стены у той клетки, где он находится:

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

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

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

НАЧАЛО

ПОКА снизу свободно

ДЕЛАТЬ вниз

ПОКА справа свободно

ДЕЛАТЬ вправо

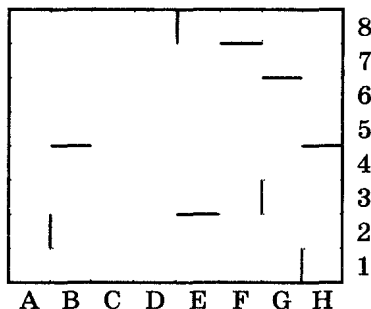
ПОКА сверху свободно

ДЕЛАТЬ вверх

ПОКА слева свободно

ДЕЛАТЬ влево

КОНЕЦ



В ответе запишите число – количество таких клеток, а далее, через запятые, их адреса (сначала идет латинская буква столбца, а затем цифра строки).

Например, нижний левый угол лабиринта имеет адрес A1.

73. /1.2.2/ Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

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

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

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

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

Цикл

ПОКА < условие > команда

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

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

НАЧАЛО

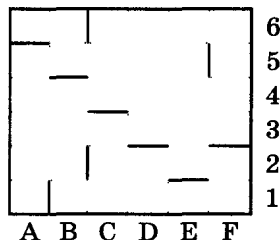
ПОКА < справа свободно > вправо

ПОКА < сверху свободно > вверх

ПОКА < слева свободно > влево

ПОКА < снизу свободно > вниз

КОНЕЦ



1) 1

2) 0

3) 3

4) 4

74. /1.2.2/ Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

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

При выполнении этой команды РОБОТ перемещается на соответствующую клетку.

Команды проверки истинности условия на наличие стены у той клетки, где он находится:

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

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

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

НАЧАЛО

ПОКА снизу свободно

ДЕЛАТЬ вниз

ПОКА справа свободно

ДЕЛАТЬ вправо

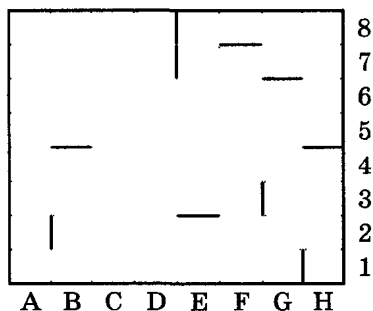
ПОКА сверху свободно

ДЕЛАТЬ вверх

ПОКА слева свободно

ДЕЛАТЬ влево

КОНЕЦ



В ответе запишите число – количество таких клеток, а далее, через запятые, их адреса (сначала идет латинская буква столбца, а затем цифра строки).

Например, нижний левый угол лабиринта имеет адрес A1.

75. /1.2.3/ Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

Бейсик	Паскаль	Алгоритмический
$a = 2599$ $b = (a \text{ MOD } 10) * 1000 + 26$ $a = (b \setminus 10) \text{ MOD } 100$ \ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a := 2599;$ $b := (a \bmod 10) * 1000 + 26;$ $a := (b \div 10) \bmod 100;$ {div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}	$a := 2599$ $b := \text{mod}(a, 10) * 1000 + 26$ $a := \text{mod}(\text{div}(b, 10), 100)$ div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

- 1) $a = 26, b = 9026$
- 2) $a = 2, b = 9026$
- 3) $a = 26, b = 9926$
- 4) $a = 2, b = 9926$

76. /1.2.3/ Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

Бейсик	Паскаль	Алгоритмический
$a = 1075$ $b = (a \text{ MOD } 1000) * 10$ $a = a + (b \setminus 100)$ '\ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a := 1075;$ $b := (a \bmod 1000) * 10;$ $a := a + (b \div 100);$ {div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}	$a := 1075$ $b := \text{mod}(a, 1000) * 10$ $a := a + \text{div}(b, 100)$ div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

- 1) $a = 1150, b = 750$
- 2) $a = 1082, b = 1070$
- 3) $a = 1082, b = 750$
- 4) $a = 1085, b = 1070$

77. /1.2.3/ Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

Бейсик	Паскаль	Алгоритмический
$a = 1819$ $b = (a \setminus 100) * 10 + 9$ $a = (10 * b - a) \text{ MOD } 100$ '\ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a := 1819;$ $b := (a \div 100) * 10 + 9;$ $a := (10 * b - a) \bmod 100;$ {div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}	$a := 1819$ $b := \text{div}(a, 100) * 10 + 9$ $a := \text{mod}(10 * b - a, 100)$ div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

- 1) $a = 81, b = 199$
- 2) $a = 81, b = 189$
- 3) $a = 71, b = 199$
- 4) $a = 71, b = 189$

78. /1.2.3/ Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

Бейсик	Паскаль	Алгоритмический
$a = 2468$ $b = (a \text{ MOD } 1000) * 10$ $a = a \setminus 1000 + b$ \ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a := 2468;$ $b := (a \bmod 1000) * 10;$ $a := a \div 1000 + b;$ {div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}	$a := 2468$ $b := \text{mod}(a, 1000) * 10$ $a := \text{div}(a, 1000) + b$ div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

- 1) $a = 22, b = 20;$
- 2) $a = 4682, b = 4680;$
- 3) $a = 8246, b = 246;$
- 4) $a = 470, b = 468.$

79. /1.2.3/ Определите значение целочисленных переменных a и b после выполнения фрагмента программы.

Бейсик	Паскаль	Алгоритмический
$a = 4 + 8 * 3$ $b = (a \text{ MOD } 10) + 15$ $a = (b \setminus 10) + 3$ \ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a := 4 + 8 * 3;$ $b := (a \bmod 10) + 15;$ $a := (b \div 10) + 3;$ {div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}	$a := 4 + 8 * 3$ $b := \text{mod}(a, 10) + 15$ $a := \text{div}(b, 10) + 3$ div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

- 1) $a = 5, b = 21$
- 2) $a = 5, b = 23$
- 3) $a = 6, b = 23$
- 4) $a = 6, b = 21$

80. /1.2.4/ Значения двух массивов $A[1..100]$ и $B[1..100]$ задаются с помощью следующего фрагмента программы:

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 100 A(n)=n-10 NEXT n FOR n=1 TO 100 B(n)=A(n)*n NEXT n	for n:=1 to 100 do A[n]:=n-10; for n:=1 to 100 do B[n]:=A[n]*n	<u>нц для n от 1 до 100</u> A[n]=n-10 <u>кц</u> <u>нц для n от 1 до 100</u> B[n]=A[n]*n <u>кц</u>

Сколько элементов массива B будут иметь положительные значения?

- 1) 10 2) 50 3) 90 4) 100

81. /1.2.4/ Значения двух массивов $A[1..200]$ и $B[1..200]$ задаются с помощью следующего фрагмента программы:

Бейсик	Паскаль	Алгоритмический
FOR n = 1 TO 200 A(n) = n + 100 NEXT n FOR n = 1 TO 200 B(n) = 2 * A(n) - 500 NEXT n	for n:=1 to 200 do A[n]:=n+100; for n:=1 to 200 do B[n]:=2*A[n]-500	<u>нц для n от 1 до 200</u> A[n]=n+100 <u>кц</u> <u>нц для n от 1 до 200</u> B[n]=2*A[n]-500 <u>кц</u>

Сколько элементов массива B будут иметь положительные значения?

- 1) 50 2) 51 3) 149 4) 150

82. /1.2.4/ Значения двух массивов $A[1..300]$ и $B[1..300]$ задаются с помощью следующего фрагмента программы:

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 300 A(n)=100-n NEXT n FOR n=1 TO 300 B(n)=2*A(n)+1 NEXT n	for n:=1 to 300 do A[n]:=100-n; for n:=1 to 300 do B[n]:=2*A[n]+1	<u>нц для n от 1 до 300</u> A[n]=100-n <u>кц</u> <u>нц для n от 1 до 300</u> B[n]=2*A[n]+1 <u>кц</u>

Сколько элементов массива B будут иметь положительные значения?

- 1) 99 2) 100 3) 200 4) 201

83. /1.2.4/ Значения двумерного массива размера 7×7 задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический
<pre>FOR n=1 TO 7 FOR k=1 TO 7 B(n, k)=k-n NEXT k NEXT n</pre>	<pre>for n:=1 to 7 do for k:=1 to 7 do B[n, k]:=k-n;</pre>	<pre>нц для n от 1 до 7 нц для k от 1 до 7 B[n, k]=k-n кц кц</pre>

Сколько элементов массива будут иметь положительные значения?

- 1) 49 2) 28 3) 21 4) 7

84. /1.2.3/ Элементы двумерного массива A размером $N \times N$ первоначально были равны 1000. Затем значения некоторых из них меняются с помощью вложенного оператора цикла в представленном фрагменте программы (ниже представлен фрагмент одной и той же программы, записанный на разных языках программирования).

Бейсик	Паскаль	Алгоритмический
<pre>k:= 0 FOR i = 1 TO N FOR j = N - i+1 TO N k = k + 1 A(i, j) = k NEXT j NEXT i</pre>	<pre>k:= 0; for i:= 1 to N do for j:= N - i + 1 to N do begin k:= k + 1; A[i, j]:= k; end</pre>	<pre>k:= 0 нц для i от 1 до N нц для j от N - i + 1 до N k:= k + 1 A[i, j]:= k кц кц</pre>

Какой элемент массива в результате будет иметь минимальное значение?

- 1) $A(1, 1)$ 2) $A(1, N)$ 3) $A(N, 1)$ 4) $A(N, N)$

85. /1.2.3/ Значения двух массивов $A[1..100]$ и $B[1..100]$ задаются с помощью следующего фрагмента программы:

Бейсик	Паскаль	Алгоритмический
<pre>FOR n = 1 TO 100 A(n) = (n - 80)*(n - 80) NEXT n FOR n = 1 TO 100 B(101-n) = A(n) NEXT n</pre>	<pre>for n:=1 to 100 do A[n]:= (n-80)*(n-80); for n:= 1 to 100 do B[101-n]:= A[n];</pre>	<pre>нц для n от 1 до 100 A[n]=(n-80)*(n-80) кц нц для n от 1 до 100 B[101-n]=A[n] кц</pre>

Какой элемент массива B будет наибольшим?

- 1) $B[1]$ 2) $B[21]$ 3) $B[80]$ 4) $B[100]$

86. /1.2.4/ Опишите на русском языке или одним из языков программирования алгоритм подсчета суммы произведений последовательных пар элементов в целочисленном массиве из 30 элементов. (Это означает, что надо сосчитать сумму произведений первого и второго, третьего и четвертого, пятого и шестого элементов и т.д.)
87. /1.2.4/ Опишите на русском языке или одним из языков программирования алгоритм поиска номера первого из двух последовательных элементов в целочисленном массиве из 30 элементов, произведение которых максимально (если таких пар несколько, то можно выбрать любую из них).
88. /1.2.4/ Опишите на русском языке или одним из языков программирования алгоритм поиска номера первого из двух последовательных элементов в целочисленном массиве из 30 элементов, сумма которых максимальна (если таких пар несколько, то можно выбрать любую из них).
89. /1.2.4/ Опишите на русском языке или одним из языков программирования алгоритм подсчета среднего значения отрицательных элементов в целочисленном массиве из 30 элементов в предположении, что в нем есть хотя бы один отрицательный элемент.
90. /1.2.4/ Опишите на русском языке или одним из языков программирования алгоритм подсчета максимального количества подряд идущих отрицательных элементов в целочисленном массиве длины 30.
91. /1.2.4/ Опишите на русском языке или одним из языков программирования алгоритм подсчета максимального количества подряд идущих элементов, каждый из которых больше предыдущего, в целочисленном массиве длины 30.

Основы логики

92. /1.3.1/ Для какого числа X истинно высказывание:

$$\neg ((X > 3) \rightarrow (X > 4))?$$

1) 1

2) 2

3) 3

4) 4

93. /1.3.1/ Для какого числа X истинно высказывание:

$$((X > 2) \vee (X < 2)) \rightarrow (X > 4)?$$

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

94. /1.3.1/ Для какого числа X истинно высказывание:

$$((X > 3) \vee (X < 3)) \rightarrow (X < 1)?$$

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

95. /1.3.2/ Для какого из указанных значений числа X истинно высказывание:

$$(X > 4) \vee ((X > 1) \rightarrow (X > 4))?$$

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

96. /1.3.1/ Для какого слова истинно высказывание:

\neg (Первая буква слова согласная \rightarrow (Вторая буква слова гласная \vee Последняя буква слова гласная))

- 1) ГОРЕ 2) ПРИВЕТ 3) КРЕСЛО 4) ЗАКОН

97. /1.3.1/ Укажите, для какого слова истинно высказывание:

(Первая буква слова гласная \vee Пятая буква слова согласная) \rightarrow Вторая буква слова гласная.

- 1) АРБУЗ 2) ОТВЕТ 3) КРЕСЛО 4) ПРИВАЛ

98. /1.3.2/ Какое логическое выражение равносильно выражению $\neg (A \wedge B) \wedge \neg C$?

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

99. /1.3.2/ Укажите, какое логическое выражение равносильно выражению $\neg (\neg A \wedge B) \vee \neg C$.

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

100. /1.3.2/ Укажите, какое логическое выражение равносильно выражению $\neg A \vee \neg (B \vee C)$.

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

101. /1.3.2/ Какое логическое выражение равносильно выражению $\neg(\neg A \vee B) \vee \neg C$?

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

2) $\neg A \vee B \vee \neg C$

3) $A \vee \neg B \vee \neg C$

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

102. /1.3.2/ Укажите, какое логическое выражение равносильно выражению $\neg(A \vee \neg B \vee C)$.

1) $\neg A \vee B \vee \neg C$

3) $\neg A \vee \neg B \vee \neg C$

2) $A \wedge \neg B \wedge C$

4) $\neg A \wedge B \wedge \neg C$

103. /1.3.2/ Каково наибольшее целое число X , при котором истинно высказывание:

$$(90 < X \cdot X) \rightarrow (X < (X-1))?$$

104. /1.3.2/ A, B, C — целые числа, для которых истинно высказывание:

$$\neg(A=B) \wedge ((A>B) \rightarrow (B>C)) \wedge ((B>A) \rightarrow (C>B))$$

Чему равно B , если $A=45, C=43$?

105. /1.3.2/ A, B, C — целые числа, для которых истинно высказывание:

$$\neg(A=B) \wedge ((B<A) \rightarrow (2C>A)) \wedge ((A<B) \rightarrow (A>2C))$$

Чему равно A , если $C=8, B=18$?

106. /1.3.2/ Каково наибольшее целое число X , при котором истинно высказывание:

$$(50 < X \cdot X) \rightarrow (50 > (X+1) \cdot (X+1))?$$

107. /1.3.2/ Укажите значения логических переменных K, L, M, N , при которых логическое выражение:

$$(K \vee M) \rightarrow (M \vee \neg L \vee N)$$

ложно.

Ответ запишите в виде строки из четырех символов: значений переменных K, L, M и N (в указанном порядке). Так, например, строка 0101 соответствует тому, что $K=0, L=1, M=0, N=1$.

108. /1.3.2/ Сколько различных решений имеет уравнение

$$((K \vee L) \wedge (M \vee N)) = 1,$$

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

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

109. /1.3.2/ Дано логическое выражение:

$$(M \vee \neg L \vee \neg N) \rightarrow (K \vee \neg L \vee \neg N).$$

Укажите значения переменных K, L, M, N , при которых логическое выражение **ложно**.

Ответ запишите в виде строки из четырех символов — значений переменных K, L, M и N (в указанном порядке). Так, например, строка 0101 соответствует тому, что $K = 0, L = 1, M = 0, N = 1$.

110. /1.3.2/ Сколько различных решений имеет уравнение

$$((K \vee L) \rightarrow (L \wedge M \wedge N)) = 0,$$

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

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

111. /1.3.2/ В школьном первенстве по настольному теннису в четверку лучших вошли девушки: Наташа, Маша, Люда и Рита. Самые горячие болельщики высказали свои предположения о распределении мест в дальнейших состязаниях.

Один считает, что первой будет Наташа, а Маша будет второй.

Другой болельщик на второе место прочит Люду, а Рита, по его мнению, займет четвертое место.

Третий любитель тенниса с ними не согласился. Он считает, что Рита займет третье место, а Наташа будет второй.

Когда соревнования закончились, оказалось, что каждый из болельщиков был прав только в одном из своих прогнозов.

Какое место на чемпионате заняли Наташа, Маша, Люда, Рита?

(В ответе перечислите подряд без пробелов числа, соответствующие местам девочек в указанном порядке имен.)

- 112. /1.3.2/** Восемь школьников, оставшихся в классе на перемене, были вызваны к директору. Один из них разбил окно в кабинете. На вопрос директора, кто это сделал, были получены следующие ответы:

Егор: «Разбил Андрей!»
Света: «Вика разбила».
Оля: «Разбила Света».
Миша: «Это кто-то с улицы!»
Надя: «Да, Оля права...»
Коля: «Это либо Вика, либо Света!»
Андрей: «Ни Вика, ни Света этого не делали».
Вика: «Андрей не бил!»

Кто разбил окно, если известно, что из этих восьми высказываний истинно ровно три?

Ответ запишите в виде первой буквы имени.

- 113. /1.3.2/** Восемь школьников, оставшихся в классе на перемене, были вызваны к директору. Один из них разбил окно в кабинете. На вопрос директора, кто это сделал, были получены следующие ответы:

Соня: «Это сделал Володя».
Миша: «Это ложь!»
Володя: «Я разбил!»
Аня: «Это я разбила!»
Оля: «Аня не разбивала!»
Рома: «Разбила либо Соня, либо Оля...»
Коля: «Девочки этого не делали».
Толя: «Коля разбил!»

Кто разбил окно, если известно, что из этих восьми высказываний истинно только два?

Ответ запишите в виде первой буквы имени.

- 114. /1.3.2/** Три школьника, Миша (*М*), Коля (*К*) и Сергей (*С*), оставшиеся в классе на перемене, были вызваны к директору по поводу разбитого в это время окна в кабинете. На вопрос директора о том, кто это сделал, мальчики ответили следующее:

Миша: «Я не бил окно, и Коля тоже...»

Коля: «Миша не разбивал окно, это Сергей разбил футбольным мячом!»

Сергей: «Я не делал этого, стекло разбил Миша».

Стало известно, что один из ребят сказал чистую правду, второй в одной части заявления соврал, а другое его высказывание истинно, а третий оба факта искажил. Зная это, директор смог докопаться до истины.

Кто разбил стекло в классе? В ответе запишите только первую букву имени.

115. /1.3.3/ Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z .

Дан фрагмент таблицы истинности выражения F :

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

Какое выражение соответствует F ?

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

116. /1.3.3/ Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z .

Дан фрагмент таблицы истинности выражения F :

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

Какое выражение соответствует F ?

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

117. /1.3.3/ Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z .

Дан фрагмент таблицы истинности выражения F :

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

Какое выражение соответствует F ?

- 1) $\neg X \vee Y \vee \neg Z$
 - 2) $X \wedge Y \wedge \neg Z$
 - 3) $\neg X \wedge \neg Y \wedge Z$
 - 4) $X \vee \neg Y \vee Z$
118. /1.3.3/ Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z .

Дан фрагмент таблицы истинности выражения F :

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

Какое выражение соответствует F ?

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

Моделирование

119. /1.4.2/ Таблица стоимости перевозок устроена следующим образом: числа, стоящие на пересечениях строк и столбцов таблиц, означают стоимость проезда между соответствующими соседними станциями. Если пересечение строки и столбца пусто, то станции не являются соседними.

Укажите таблицу, для которой выполняется условие: «Минимальная стоимость проезда из A в B не больше 6».

Стоимость проезда по маршруту складывается из стоимостей проезда между соответствующими соседними станциями.

1)

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

2)

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

3)

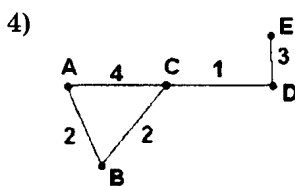
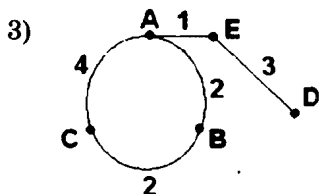
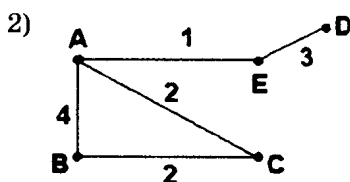
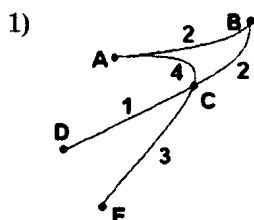
	A	B	C	D	E
A			3	1	
B			4		1
C	3	4			2
D	1				
E		1	2		

4)

	A	B	C	D	E
A				1	
B			4		1
C		4		4	2
D	1				
E		1	2		

120./1.4.2/ В таблице приведена стоимость перевозок между соседними железнодорожными станциями. Укажите схему, соответствующую таблице.

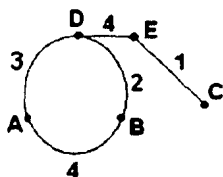
	A	B	C	D	E
A		2	4		1
B	2		2		
C	4	2			
D					3
E	1			3	



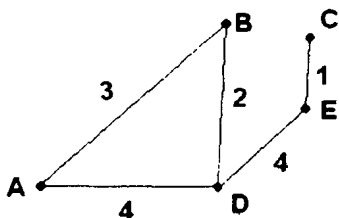
121. /1.4.2/ В таблице приведена стоимость перевозок между соседними железнодорожными станциями. Укажите схему, соответствующую таблице.

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

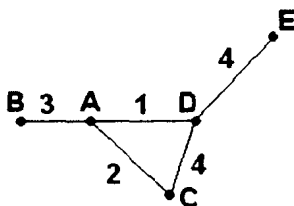
1)



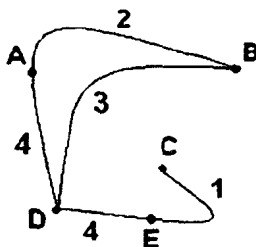
2)



3)



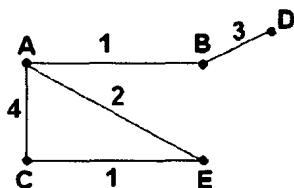
4)



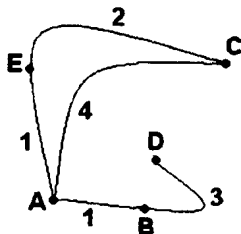
122. /1.4.2/ В таблице приведена стоимость перевозок между соседними железнодорожными станциями. Укажите схему, соответствующую таблице.

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

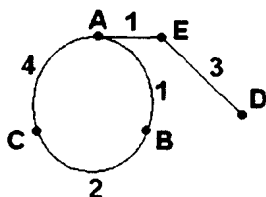
1)



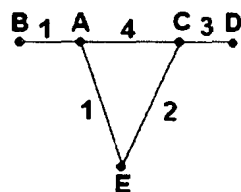
2)



3)



4)



РАЗДЕЛ 2. ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

Программные средства информационных и коммуникационных технологий

123. /2.2.3/ В некотором каталоге хранился файл **Пушкин.doc**. После того как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл **Пушкин.doc**, полное имя файла стало **F:\Литература\Поэты\XIX\Пушкин.doc**. Каково имя вновь созданного каталога?

1) Поэты

3) XIX

2) F:\Литература

4) Литература

124. /2.2.3/ Перемещаясь из одного каталога в другой, пользователь последовательно посетил каталоги **DOC**, **USER**, **SCHOOL**, **A:**, **LETTER**, **INBOX**. Каково полное имя каталога, в котором оказался пользователь?

Примечание: при каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше.

1) INBOX

3) A:\SCHOOL\USER\DOC

2) A:\ LETTER\INBOX

4) LETTER\INBOX

125. /2.2.3/ Перемещаясь из одного каталога в другой, пользователь последовательно посетил каталоги **DOC, USER, SCHOOL, A:\, LETTER, INBOX. Каково полное имя каталога, из которого начал перемещение пользователь?**

Примечание: при каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше.

- 1) **A:\DOC**
- 2) **A:\ LETTER\INBOX**
- 3) **A:\SCHOOL\USER\DOC**
- 4) **A:\DOC\USER\SCHOOL**

126. /2.2.3/ В некотором каталоге хранился файл **Дневник.txt. После того как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл **Дневник.txt**, полное имя файла стало **A:\SCHOOL\USER\TXT\MAY\Дневник.txt**. Каково полное имя каталога, в котором хранился файл до перемещения?**

- 1) **MAY**
- 2) **A:\SCHOOL\USER\TXT**
- 3) **TXT**
- 4) **A:\SCHOOL\USER\TXT\MAY**

127. /2.3.3/ Находясь в корневом каталоге только что отформатированного диска, ученик создал 3 каталога. Затем в каждом из созданных каталогов он создал еще по 4 каталога. Сколько всего оказалось на диске каталогов, включая корневой?

- 1) 12
- 2) 13
- 3) 15
- 4) 16

128. /2.3.3/ Файл, полное имя которого было **C:\doc\MyLetter.C, сохранили в подкаталоге **txt** корневого каталога диска **E**. Каково полное имя сохраненного файла?**

- 1) **E:\txt\MyLetter.E**
- 2) **E:\txt\MyLetter.C**
- 3) **E:\txt\doc\MyLetter.C**
- 4) **E:\doc\txt\MyLetter.C**

Обработка графической информации

- 129.** /2.4.1/ Для хранения растрового изображения размером 64×32 пикселя отвели 1 Кбайт памяти. Каково максимально возможное число цветов в палитре изображения?
- 1) 16 2) 32 3) 64 4) 1024
- 130.** /2.4.1/ Для хранения растрового изображения размером 32×64 пикселя отвели 512 байт памяти. Каково максимально возможное число цветов в палитре изображения?
- 1) 2048 2) 1024 3) 16 4) 4
- 131.** /2.4.1/ Цвет пикселя, формируемого принтером, определяется тремя составляющими: голубой, пурпурной и желтой. Под каждую составляющую одного пикселя отвели по четыре бита. В какое количество цветов можно раскрасить пиксель?
- 1) 12 2) 122 3) 212 4) 312
- 132.** /2.4.1/ Для хранения растрового изображения размером 32×32 пикселя отвели 1 килобайт памяти. Каково максимально возможное число цветов в палитре изображения?
- 1) 16 2) 32 3) 256 4) 1024
- 133.** /2.4.1/ Цвет пикселя монитора определяется тремя составляющими: зеленой, синей и красной. Под красную и синюю составляющие одного пикселя отвели по 5 бит. Сколько бит отвели под зеленую составляющую одного пикселя, если растровое изображение размером 8×8 пикселей занимает 128 байт памяти?
- 1) 5 2) 6 3) 8 4) 16
- 134.** /2.4.1/ Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 64×64 пикселя, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно.
- 1) 128 2) 2 3) 256 4) 4
- 135.** /2.4.1/ Укажите минимальный объем памяти (в байтах), достаточный для хранения любого растрового изображения размером 8×32 пикселей, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно.
- 1) 128 2) 256 3) 512 4) 1024

Обработка информации в электронных таблицах

136. /2.5.2/ В ячейке B1 записана формула $=2*\$A1$. Какой вид приобретет формула после того, как ячейку B1 скопируют в ячейку C2?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) $=2*\$B1$ 2) $=2*\$A2$ 3) $=3*\$A2$ 4) $=3*\$B2$

137. /2.5.2/ В ячейке A1 записана формула $=\$C\$5+E4$. Какой вид приобретет формула после того, как ячейку A1 скопируют в ячейку B3?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) $=\$C\$5+F6$ 3) $=\$A\$3+E4$
2) $=SC\$5+D2$ 4) $=\$D\$7+E4$

138. /2.5.2/ В ячейке B3 записана формула $=C\$2+\$D3+2$. Какой вид приобретет формула после того, как ячейку B3 скопируют в ячейку B2?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) $=B\$2+\$D3+2$ 2) $=C\$1+\$D2+2$
3) $=C\$2+\$D2+2$ 4) $=B\$2+\$D2+2$

139. /2.5.2/ При работе с электронной таблицей в ячейке A1 записана формула $=D1-\$D2$. Какой вид приобретет формула после того, как ячейку A1 скопируют в ячейку B1?

Примечание: символ \$ в формуле обозначает абсолютную адресацию.

- 1) $=E1-\$E2$ 2) $=E1-\$D2$ 3) $=E2-\$D2$ 4) $=D1-\$E2$

140. /2.5.2/ В ячейке B3 записана формула $=C\$2+\$D3+2$. Какой вид приобретет формула после того, как ячейку B3 скопируют в ячейку B2?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) $=B\$2+\$D3+2$ 3) $=C\$2+\$D2+2$
2) $=C\$1+\$D2+2$ 4) $=B\$2+\$D2+2$

141. /2.5.2/ Дан фрагмент электронной таблицы:

	A	B	C
1	4	2	
2	1	3	$=A2 + 2*B\$2$

Чему станет равным значение ячейки C1, если в нее скопировать формулу из ячейки C2?

Знак \$ обозначает абсолютную адресацию.

- 1) 5 2) 8 3) 10 4) 11

142. /2.5.2/ Дан фрагмент электронной таблицы:

	A	B	C
1	10	20	$=A1+B\$1$
2	30	40	

Чему станет равным значение ячейки C2, если в нее скопировать формулу из ячейки C1?

Знак \$ обозначает абсолютную адресацию.

- 1) 40 2) 50 3) 60 4) 70

143. /2.5.2/ Дан фрагмент электронной таблицы:

	A	B	C
1	5	6	
2	4	2	$=\$A2+B2$

Чему станет равным значение ячейки C1, если в нее скопировать формулу из ячейки C2?

Знак \$ обозначает абсолютную адресацию.

- 1) 6 2) 7 3) 8 4) 11

144. /2.5.2/ Дан фрагмент электронной таблицы:

	A	B	C
1	10	20	$=A\$1+B\1
2	30	40	

Чему станет равным значение ячейки C2, если в нее скопировать формулу из ячейки C1?

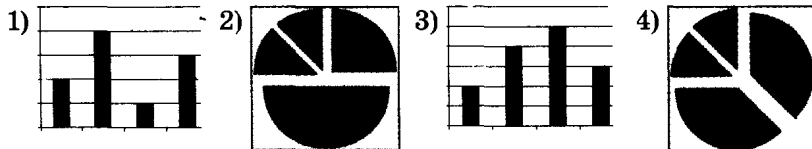
Знак \$ обозначает абсолютную адресацию.

- 1) 30 2) 40 3) 50 4) 60

145. /2.5.3/ Дан фрагмент электронной таблицы:

	A	B
1	$=B1+1$	1
2	$=A1+2$	2
3	$=B2-1$	
4	$=A3$	

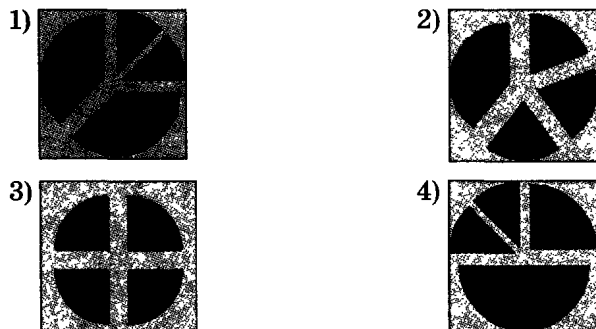
После выполнения вычислений была построена диаграмма по значениям диапазона ячеек A1:A4. Укажите получившуюся диаграмму.



146. /2.5.3/ Дан фрагмент электронной таблицы:

	A	B	C	D
1	$=C2$	$=C1-A1$	$=A1*2$	$=B1*2+B2$
2		4	2	

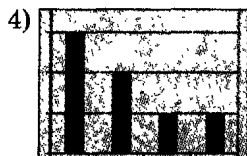
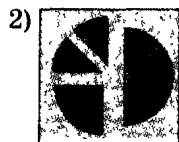
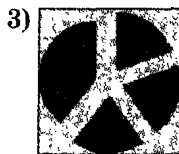
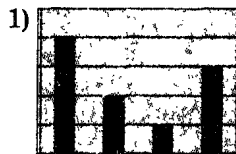
После выполнения вычислений была построена диаграмма по значениям диапазона ячеек A1:D1. Укажите получившуюся диаграмму.



147. /2.5.3/ Дан фрагмент электронной таблицы:

	A	B	C	D
1	$=B2 + C2$	$=C1 + B2$	$=A1 - C2$	$= B1-C1$
2		1	3	

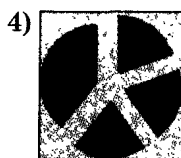
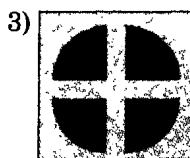
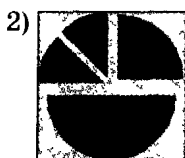
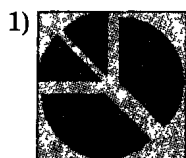
После выполнения вычислений была построена диаграмма по значениям диапазона ячеек A1:D1. Укажите получившуюся диаграмму.



148. /2.5.3/ Дан фрагмент электронной таблицы:

	A	B
1		$=B3-A2$
2	2	$=A2+B1$
3		3
4		$=B2-A2$

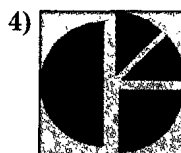
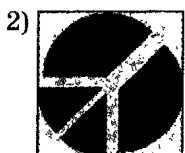
После выполнения вычислений была построена диаграмма по значениям диапазона ячеек B1:B4. Укажите получившуюся диаграмму.



149. /2.5.3/ Дан фрагмент электронной таблицы:

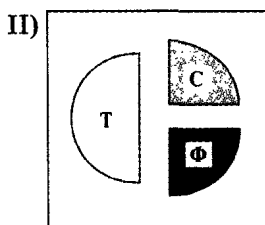
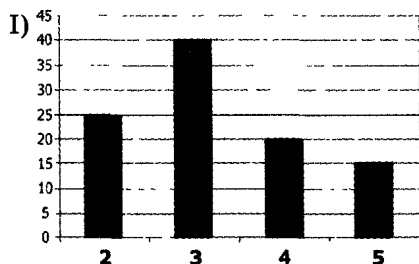
	A	B	C	D
1		3	4	
2	$=C1-B1$	$=B1-A2*2$	$=C1/2$	$=B1+B2$

После выполнения вычислений была построена диаграмма по значениям диапазона ячеек A2:D2. Укажите получившуюся диаграмму.



150. /2.5.3/ В цехе трудятся рабочие трех специальностей — токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд, не меньший второго и не больший пятого. На диаграмме I отражено количество рабочих с разными разрядами, а на диаграмме II — распределение рабочих по специальностям.

Каждый рабочий имеет только одну специальность и один разряд.



Имеются четыре утверждения:

А) Все рабочие третьего разряда могут быть токарями.

Б) Все рабочие третьего разряда могут быть фрезеровщиками.

В) Все слесари могут быть пятого разряда.

Г) Все токари могут быть четвертого разряда.

Какое из этих утверждений следует из анализа обеих диаграмм?

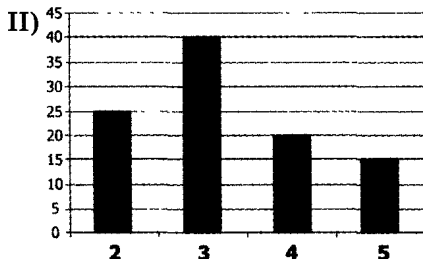
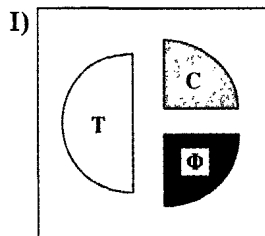
1) А

2) Б

3) В

4) Г

151. /2.5.3/ В цехе трудятся рабочие трех специальностей — токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд, не меньший второго и не больший пятого. На диаграмме I отражено распределение рабочих по специальностям, а на диаграмме II — количество рабочих с различными разрядами. Каждый рабочий может иметь только одну специальность и один разряд.



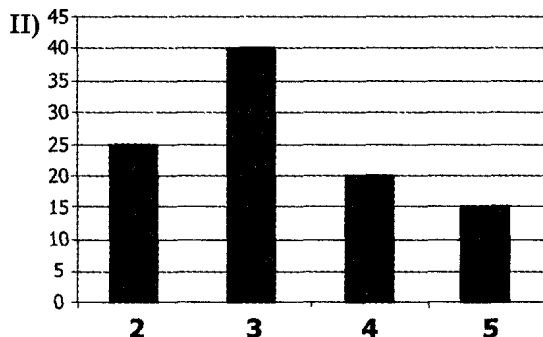
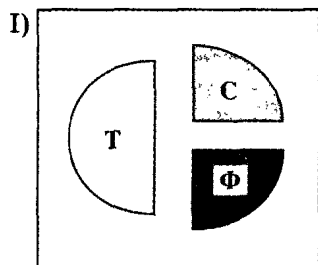
Имеются четыре утверждения:

- А) Среди слесарей найдется хотя бы один третьего разряда.
- Б) Среди токарей найдется хотя бы один второго разряда.
- В) Все токари могут иметь четвертый разряд.
- Г) Все фрезеровщики могут иметь третий разряд.

Какое из этих утверждений следует из анализа обеих диаграмм?

- 1) А 2) Б 3) В 4) Г

152. /2.5.3/ В цехе трудятся рабочие трех специальностей — токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд, не меньший второго и не больший пятого. На диаграмме I отражено распределение рабочих по специальностям, а на диаграмме II — количество рабочих с разными разрядами. Каждый рабочий может иметь только одну специальность и один разряд.



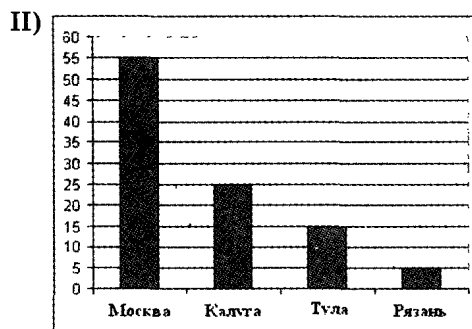
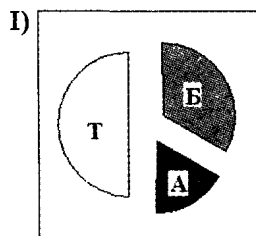
Имеются четыре утверждения:

- А) Все рабочие третьего разряда могут быть токарями.
- Б) Все токари могут быть третьего разряда.
- В) Все слесари могут быть пятого разряда.
- Г) Все фрезеровщики могут быть четвертого разряда.

Какое из этих утверждений следует из анализа обеих диаграмм?

- 1) А 2) Б 3) В 4) Г

153. /2.5.3/ Завод выпускает бульдозеры (Б), автокраны (А) и тракторы (Т). Он поставляет их в Москву, Калугу, Тулу и Рязань. На диаграмме I показано распределение выпускаемой продукции по видам. На диаграмме II показано, сколько единиц техники должно быть продано в каждый регион.



Имеются четыре утверждения:

А) Возможно, что вся техника, проданная в Калугу, — бульдозеры.

Б) Среди техники, проданной в Москву, нет ни одного трактора.

В) В Калугу продано больше тракторов, чем бульдозеров.

Г) В Москву продано больше бульдозеров, чем автокранов.

Какое из этих утверждений следует из анализа обеих диаграмм?

- 1) А 2) Б 3) В 4) Г

Хранение, поиск и сортировка информации в базах данных

154. /2.6.1/ Количество полей в базе данных структуры, представленной таблицей:

ФИО	Класс	Город	Школа	Оценка
Петров П.П.	9	Москва	15	4
Иванов И.И.	10	Сочи	16	5

равно:

- 1) 10 2) 2 3) 3 4) 5

155. /2.6.3/ Из правил соревнования по тяжелой атлетике:

Тяжелая атлетика — это прямое соревнование, когда каждый атлет имеет три попытки в рывке и три попытки в толчке. Самый тяжелый вес поднятой штанги в каждом упражнении суммируется в общем зачете. Если спортсмен потерпел неудачу во всех трех попытках в рывке, он может продолжить соревнование в толчке, но уже не сможет занять какое-либо место по сумме 2-х упражнений. Если два спортсмена заканчивают состязание с одинаковым итоговым результатом, высшее место присуждается спортсмену с меньшим весом. Если же вес спортсменов одинаков, преимущество отдается тому, кто первым поднял победный вес.

Таблица результатов соревнований по тяжелой атлетике:

Фамилия, И.О.	Вес спортсмена	Взято в рывке	Рывок с попытки	Взято в толчке	Толчок с попытки
Айвазян Г.С.	77,1	147,5	3	200,0	2
Викторов М.П.	79,1	150,0	1	202,5	1
Гордезиани Б.Ш.	78,2	150,0	2	200,0	1
Михальчук М.С.	78,2	150,0	1	202,5	3
Пай С.В.	79,5	—	—	205,0	1
Шапсугов М.Х.	77,1	150,0	3	197,5	1

Кто победил в рывке?

1) Викторов М.П.

3) Михальчук М.С.

2) Гордезиани Б.Ш.

4) Шапсугов М.Х.

157. (2.6.4) База данных «Учащиеся» содержит поля:

Поле	Тип	Примечание
Фамилия	текст	
Имя	текст	
Пол	текст	м или д
Год рождения	число	
Рост	число	
Вес	число	
Увлечение	текст	

Как следует записать условие отбора при фильтрации, которое позволит сформировать список девушек для участия в конкурсе красоты в декабре 2005 г.? (Отбираются девушки, имеющие рост более 180 см и возраст не менее 16 лет.)

- 1) Пол="д" ИЛИ Рост > 180 И Год рождения = 1989.
- 2) Пол="д" И Рост > 180 ИЛИ Год рождения < 1989.
- 3) Пол="д" И Рост > 180 И Год рождения < 1990.
- 4) Пол="д" И Рост = 180 И Год рождения < 1989.

158. /2.6.3/ Далее в табличной форме представлен фрагмент базы данных о результатах тестирования учащихся (используется столбчатая шкала):

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Аганян	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григорчук	м	54	74	68	75	83
Роднина	ж	71	63	56	82	79
Сергеенко	ж	33	25	74	38	46
Черепанова	ж	18	92	83	28	61

Сколько записей в данном фрагменте удовлетворяют условию:

«Пол= 'м' ИЛИ Химия > Биология»?

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

159. /2.6.3/ Вот фрагмент таблицы результатов тестирования:

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Аганян	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григорчук	м	54	74	68	75	83
Роднина	ж	71	63	56	82	79
Сергеенко	ж	33	25	74	38	46
Черепанова	ж	18	92	83	28	61

Сколько записей в этом фрагменте удовлетворяют условию «Пол='ж' ИЛИ Математика + Информатика > 120»?

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

160. /2.6.3/ Сколько записей в нижеследующем фрагменте турнирной таблицы удовлетворяют условию «Место=4 ИЛИ (В>4 И О>=5)»?

Место	Участник	В	Н	П	О		
1	Силин	5	3	1	6		
2	Клеменс	6	0	3	6		
3	Холево	5	1	4	5		
4	Яшвили	3	5	1	5		
5	Бергер	3	3	3	4		
6	Численко	3	2	4	4		

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

Телекоммуникационные технологии

161. /2.7.1/ Доступ к файлу ftp.net, находящемуся на сервере txt.org, осуществляется по протоколу http. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

А	Б	В	Г	Д	Е	Ж
.net	ftp	://	http	/	.org	txt

162. /2.7.1/ Доступ к файлу http.txt, 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.

А	Б	В	Г	Д	Е	Ж
://	.net	.txt	http	ftp	/	www

163. /2.7.1/ Доступ к файлу uk.net, находящемуся на сервере org.de, осуществляется по протоколу ftp. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

А	Б	В	Г	Д	Е	Ж
org	uk	.de	.net	://	ftp	/

164. /2.7.1/ Доступ к файлу `www.txt`, находящемуся на сервере `ftp.net`, осуществляется по протоколу `http`. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла.

А	Б	В	Г	Д	Е	Ж
.txt	http	/	://	.net	www	ftp

165. /2.7.3/ В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке **возрастания** количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции “ИЛИ” в запросе используется символ `|`, а для логической операции “И” — `&`.

А	волейбол	баскетбол	подача	
Б	волейбол	баскетбол	подача	блок
В	волейбол	баскетбол		
Г	волейбол & баскетбол & подача			

166. /2.7.3/ В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке **убывания** количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции “ИЛИ” в запросе используется символ `|`, а для логической операции “И” — `&`.

А	продажа	принтеры		
Б	сервис	продажа	принтеры	сканеры
В	принтеры & сканеры & продажа & сервис			
Г	(продажа принтеры) & (сервис сканеры)			

167. /2.7.3/ В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке **возрастания** количества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции “ИЛИ” в запросе используется символ `|`, а для логической операции “И” — `&`.

А	(принтеры & сканеры) продажа	
Б	принтеры & сканеры & продажа & сервис	
В	принтеры & продажа	
Г	сервис	принтеры сканеры

168. /2.7.3/ В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке **возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.**

Для обозначения логической операции “ИЛИ” в запросе используется символ $|$, а для логической операции “И” — символ $\&$.

А	разведение $\&$ содержание $\&$ меченосцы $\&$ сомики
Б	содержание $\&$ меченосцы
В	(содержание $\&$ меченосцы) $ $ сомики
Г	содержание $\&$ меченосцы $\&$ сомики

Технологии программирования

169. /2.8.1/ Требовалось написать программу, которая решает неравенство $ax+b>0$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и неправильно написал программу.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b,x); if a = 0 then write('любое число') else if a > 0 then write('x > ', -b/a) else write('x < ', -b/a); end.</pre>	<pre>INPUT a, b, x IF a = 0 THEN PRINT «любое число» ELSE IF a > 0 THEN PRINT «x > », -b/a ELSE PRINT «x < », -b/a END</pre>	<pre>void main(void) { float a,b,x; scanf(«%f%f%f», &a,&b,&x); if (a==0) printf(«любое число»); else if (a>0) printf(«x>%f», -b/a); else printf(«x<%f», -b/a); }</pre>

Последовательно выполните три задания:

- 1) Приведите пример таких чисел a, b, x , при которых программа неверно решает поставленную задачу.
- 2) Укажите, какая часть программы является лишней.

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

170. /2.8.1/ Требовалось написать программу, которая решает неравенство $ax + b > 0$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и неправильно написал программу.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b,x); if a = 0 then if b > 0 then write('любое число') else write('нет решений') else write('x > ', -b/a); end.</pre>	<pre>INPUT a, b, x IF a = 0 THEN IF b > 0 THEN PRINT «любое число» ELSE PRINT «нет решений» ENDIF ELSE PRINT «x > », -b/a ENDIF END</pre>	<pre>void main(void) { float a,b,x; scanf(«%f%f%f», &a,&b,&x); if (a==0) if (b > 0) printf(«любое число»); else printf(«нет решений»); else printf(«x > %f», -b/a); }</pre>

Последовательно выполните три задания:

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

171. /2.8.1/ Требовалось написать программу, которая решает уравнение $ax + b = 0$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и неправильно написал программу.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a, b, x: real; begin readln(a,b,x); if b = 0 then write('x = 0') else if a = 0 then write('нет решений') else write('x = ', -b/a); end.</pre>	<pre>INPUT a, b, x IF b = 0 THEN PRINT «x = 0» ELSE IF a = 0 THEN PRINT «нет реше- ний» ELSE PRINT «x = », -b/a END</pre>	<pre>void main(void) { float a,b,x; scanf(«%f%f%f», &a, &b, &x); if (b==0) printf(«x=0»); else if (a==0) printf(«нет решений»); else printf(«x=%f», -b/a); }</pre>

Последовательно выполните три задания:

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

172. /2.8.1/ Требовалось написать программу, которая решает уравнение $a|x| = b$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a, b, x: real; begin readln(a,b,x); if a = 0 then if b = 0 then write('любое число') else write('нет решений') else if b = 0 then write('x = 0') else write('x = ', b/a, или x = ', -b/a); end.</pre>	<pre>INPUT a, b, x IF a = 0 THEN IF b = 0 THEN PRINT «любое число» ELSE PRINT «нет решений» ENDIF ELSE IF b = 0 THEN PRINT «x = 0» ELSE PRINT «x = », b/a, « или x = », -b/a END IF END IF END</pre>	<pre>void main(void) {float a,b,x; scanf(«%f%f%f», &a, &b, &x); if (a==0) if (b==0) printf(«любое число»); else printf («нет решений»); else if (b==0) printf(«x = 0»); else printf(«x=%f или x=%f», b/a, -b/a); }</pre>

Последовательно выполните три задания:

1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу.

2) Укажите, какая часть программы является лишней.

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

173. /2.8.1/ Требовалось написать программу, которая решает неравенство $\langle (x-a)/(bx) > 0 \rangle$ относительно x для любых ненулевых чисел a и b ($b \neq 0$, $a \neq 0$), введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

Программа на паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b,x); if b>0 then write ('x>',a, или x<0') else if a > 0 then write('0 < x <', a) else write(a,'< x <0'); end.</pre>	<pre>INPUT a, b, x IF b > 0 THEN PRINT «x>»,a, «или x<0» ELSE IF a > 0 THEN PRINT «0<x<»,a ELSE PRINT a,«<x<0» END IF END IF END</pre>	<pre>void main(void) { float a,b,x; scanf(«%f%f%f», &a,&b,&x); if (b>0) printf(«x>%f или x<0», a); else if (a>0) printf(«0<x<%f»,a); else printf(«%f<x<0»,a); }</pre>

Последовательно выполните три задания:

1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу.

2) Укажите, какая часть программы является лишней.

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

173. Требовалось написать программу, которая решает уравнение $\langle a|x|=b \rangle$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b,x); if a = 0 then if b = 0 then write ('любое число') else write ('нет решений') else if b = 0 then write('x = 0') else write('x =',b/a,' или x =',-b/a); end.</pre>	<pre>INPUT a, b, x IF a = 0 THEN IF b = 0 THEN PRINT «любое число» ELSE PRINT «нет решений» ENDIF ELSE IF b = 0 THEN PRINT «x = 0» ELSE PRINT «x =»,b/a,» или x =», -b/a END IF END IF END</pre>	<pre>void main(void) {float a,b,x; scanf("%f%f%f", &a,&b,&x); if (a==0) if (b==0) printf(«любое число»); else printf(«нет решений»); else if (b==0) printf(«x = 0»); printf(«x=%f или x=%f», b/a,-b/a); }</pre>

Последовательно выполните три задания:

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

174. /2.8.1/ Требовалось написать программу, которая решает неравенство $\frac{(x+a)}{(bx)} < 0$ относительно x для любого ненулевого числа b и любого неотрицательного числа a и ($a \geq 0$, $b \neq 0$), введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b,x); if a = 0 then if b > 0 then write ('нет решений') else write('x > 0 или x < 0') else write(-a,'< x < 0'); end.</pre>	<pre>INPUT a, b, x IF a = 0 THEN IF b > 0 THEN PRINT «нет решений» ELSE PRINT «x>0 или x<0» END IF ELSE PRINT -a,«<x<0» END IF END</pre>	<pre>void main(void) { float a,b,x; scanf("%f%f%f", &a,&b,&x); if (a==0) if (b>0) printf(«нет решений»); else printf(«x>0 или x<0»); else printf(«%f<x<0», -a); }</pre>

Последовательно выполните три задания:

1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу.

2) Укажите, какая часть программы является лишней.

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

175. /2.8.1/ Требовалось написать программу, которая решает уравнение $|a|x| = b$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b,x); if a = 0 then if b = 0 then write('любое число') else write('нет решений') else if b = 0 then write('x = 0') else write('x =',b/a, или x =',-b/a); end.</pre>	<pre>INPUT a, b, x IF a = 0 THEN IF b = 0 THEN PRINT «любое число» ELSE PRINT «нет решений» ENDIF ELSE IF b = 0 THEN PRINT «x = 0» ELSE PRINT «x =»,b/a, » или x =«, -b/a END IF END IF END</pre>	<pre>void main(void) {float a,b,x; scanf(«%f%f%f», &a,&b,&x); if (a==0) if (b==0) printf(«любое число»); else printf(«нет решений»); else if (b==0) printf(«x = 0»); else printf(«x=%f или x=%f», b/a,-b/a); }</pre>

Последовательно выполните три задания:

1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу.

2) Укажите, какая часть программы является лишней.

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

176. /2.8.3/ На вход программе подаются сведения о сдаче экзаменов учениками 9-х классов некоторой средней школы. В первой строке сообщается количество учеников N , которое не меньше 10, но не превосходит 100, каждая из следующих N строк имеет следующий формат: <Фамилия> <Имя> <оценки>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 15 символов <оценки> — через пробел три целых числа, соответствующие оценкам по пятибалльной системе. <Фамилия> и <Имя>, а также <Имя> и <оценки> разделены одним пробелом. Пример входной строки:

Иванов Петр 4 5 4

Требуется написать программу, которая будет выводить на экран фамилии и имена трех лучших по среднему баллу учеников. Если среди остальных есть ученики, набравшие тот же средний балл, что и один из трех лучших, то следует вывести и их фамилии и имена. Требуемые имена и фамилии можно вывести в произвольном порядке.

177. /2.8.3/ На вход программе подаются сведения о пассажирах, сдавших свой багаж в камеру хранения. В первой строке задано текущее время: через двоеточие два целых числа, соответствующие часам (от 00 до 21 — ровно 2 символа) и минутам (от 00 до 59 — ровно 2 символа). Во второй строке сообщается количество пассажиров N , которое не меньше 10, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат:

<Фамилия> <время освобождения ячейки> ,

где <Фамилия> — строка, состоящая не более чем из 20 символов, <время освобождения ячейки> — через двоеточие два целых числа, соответствующие часам (от 00 до 23 — ровно 2 символа) и минутам (от 00 до 59 — ровно 2 символа). <Фамилия> и <время освобождения ячейки> разделены одним пробелом. Сведения отсортированы в порядке времени сдачи багажа.

Требуется написать программу, выводящую фамилии пассажиров, которые в ближайшие 2 ч должны освободить ячейки, в хронологическом порядке освобождения ячеек.

Пример входных данных:

10:00

3

Иванов 12:00

Петров 10:00

Сидоров 12:12

Результат работы программы для этого примера

Петров

Иванов

178. /2.8.3/ На вход программе подаются строчные английские буквы. Ввод этих символов заканчивается точкой (другие символы, отличные от “.” и букв “a”..“z”, во входных данных отсутствуют; в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введена точка). Требуется написать эффективную программу, которая будет печатать буквы, встречающиеся во входной последовательности, в порядке уменьшения частоты их встречаемости. Каждая буква должна быть распечатана один раз. Точка при этом не учитывается.

Если какие-то буквы встречаются одинаковое число раз, то они выводятся в алфавитном порядке. Например, пусть на вход подаются следующие символы:

batat.

В данном случае программа должна вывести:

atb

179. /2.8.3/ На вход программе подаются сведения об учениках некоторой средней школы. В первой строке сообщается количество учеников N , каждая из следующих N строк имеет следующий формат:

<Фамилия> <Имя> <класс>,

где <Фамилия> — строка, состоящая не более, чем из 20 символов,

<Имя> — строка, состоящая не более, чем из 15 символов, <класс> — год обучения (от 1 до 12) и заглавная буква (от “А” до “Я”) без пробела. <Фамилия> и <Имя>, а также <Имя> <класс> разделены одним пробелом. Пример входной строки:

Иванов Петр 10Б

Требуется написать программу на языке Паскаль или Бейсик, которая будет выводить на экран информацию о параллелях (годе обучения) с наибольшим числом учеников. Программа должна выводить на экран в первой строке количество

учеников в искомых параллелях, а во второй строке — в порядке возрастания номера этих параллелей через пробел. Например:

100

1 7 11

- 180. /2.8.3/** На вход программе подается текст заклинания, состоящего не более чем из 200 символов, заканчивающийся точкой (символ «точка» во входных данных единственный). Оно было зашифровано Гарри Поттером следующим образом. Сначала Гарри определил количество букв в самом коротком слове, обозначив полученное число K (словом называется непрерывная последовательность английских букв, слова друг от друга отделяются любыми другими символами, длина слова не превышает 20 символов). Затем он заменил каждую английскую букву в заклинании на букву, стоящую в алфавите на K букв ранее (алфавит считается циклическим, то есть перед буквой A стоит буква Z), оставив другие символы неизменными. Строчные буквы при этом остались строчными, а прописные — прописными. Требуется написать программу на языке Паскаль или Бейсик, которая будет выводить на экран текст расшифрованного заклинания. Например, если зашифрованный текст был таким:

Zb Ra Ca Dab Ra.

то результат расшифровки должен быть следующим:

Bl Tc Ec Fed Tc.

- 181. /2.8.3/** На вход программе подаются 365 строк, которые содержат информацию о среднесуточной температуре всех дней 2005 года. Формат каждой из строк следующий: сначала записана дата в виде dd.mm (на запись номера дня и номера месяца в числовом формате отводится строго два символа, день от месяца отделен точкой), затем через пробел (для Бейсика — через запятую) записано значение температуры — число со знаком плюс или минус, с точностью до 1 цифры после десятичной точки. Данная информация отсортирована по значению температуры, то есть хронологический порядок нарушен. Требуется написать эффективную программу на языке Паскаль или Бейсик, которая будет выводить на экран информацию о месяцах с максимальной среднемесячной температурой. Найденные максимальные значения следует выводить в отдельной строке для каждого

месяца в виде: номер месяца, значение среднемесячной температуры, округленное до одной цифры после десятичной точки.

- 182. /2.8.3/** На вход программе подаются строчные английские буквы. Ввод этих символов заканчивается точкой (другие символы, отличные от “.” и букв “a”..“z”, во входных данных отсутствуют; в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введена точка). Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая будет печатать буквы, встречающиеся во входной последовательности, в порядке увеличения частоты их встречаемости. Каждая буква должна быть распечатана один раз. Точка при этом не учитывается.

Если какие-то буквы встречаются одинаковое число раз, то они выводятся в алфавитном порядке. Например, пусть на вход подаются следующие символы:

baobaba.

В данном случае программа должна вывести:

oab

- 183. /2.8.3/** На вход программе подаются сведения о сдаче экзаменов учениками 9-х классов некоторой средней школы. В первой строке сообщается количество учеников N , которое не меньше 10, но не превосходит 100, каждая из следующих N строк имеет следующий формат:

<Фамилия> <Имя> <оценки>,

где <Фамилия> – строка, состоящая не более чем из 20 символов, <Имя> – строка, состоящая не более чем из 15 символов, <оценки> – через пробел три целых числа, соответствующие оценкам по пятибалльной системе. <Фамилия> и <Имя>, а также <Имя> и <оценки> разделены одним пробелом. Пример входной строки:

Иванов Петр 4 5 3

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая будет выводить на экран фамилии и имена трех худших по среднему баллу учеников. Если среди остальных есть ученики, набравшие тот же средний балл, что и один из трех худших, то следует вывести и их фамилии и имена.

РАЗДЕЛ 1. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

Информация и ее кодирование

Проверке знаний и умений по этой теме в вариантах экзамена 2007 г., как и в предыдущем году, было посвящено 8 заданий, из которых 6 — с выбором ответа и 2 — с кратким ответом. 5 заданий относятся к базовому уровню сложности, 5 — к повышенному. Средний процент выполнения колеблется от 88% (задание A1, проверяется знание кодовых таблиц) до 55% (задание B1 на знание математических основ записи чисел в позиционных системах счисления). Помимо задания A1, не вызвали затруднений задания A4 и A13 на двоичное кодирование — средний процент выполнения от 88% до 79%, в зависимости от варианта. Из заданий базового уровня сложности сравнительно больше затруднений вызвало задание A2 на применение алфавитного подхода к измерению количества информации — средний процент выполнения 57%. Задания повышенного уровня сложности в среднем выполнили от 55% до 72% экзаменующихся. Наиболее сложным оказалось задание B1 на знание математических основ записи чисел в позиционных системах счисления. Это задание предполагает применение знаний в новой для абитуриента ситуации, так как задач подобного типа в школе обычно не решают. Кроме того, результат выполнения этого задания тесно связан с общей математической культурой экзаменуемого. Наиболее часто встречающейся ошибкой является перечисление не всех чисел, отвечающих заданным требованиям. В 2006 г. наибольшее затруднение вызвало задание B5 на определение пропускной способности канала связи. В 2007 г. с ним справились в среднем 63% экзаменуемых, в то время как в 2006 г. — только 46%. Задание повышенного уровня A3, проверяющее умения подсчитывать информационный объем сообщения, выполнили в 2007 г. 56% экзаменовавшихся (в 2006 г. — 48%).

Рассмотрим более подробно типичные ошибки учащихся, выявленные при апробации типовых заданий.

Причиной немногочисленных ошибок при выполнении заданий типа № 1—3 на оценку информационного объема фразы в

различных кодировках обычно являются смешивание или неправильная интерпретация учащимися таких элементарных понятий, как «бит» и «байт», а также неверные арифметические вычисления. Следует также обратить внимание на то, что в ответах используются обе единицы измерения количества информации.

При выполнении этого задания у учащихся иногда возникают вопросы: «Как точно узнать количество пробелов в фразе? Считать ли точку в конце частью задания или частью оцениваемой фразы?» В точном подсчете символов в данном случае нет необходимости, поскольку в задании требуется *оценить* информационные фразы, т.е. из предложенных вариантов ответа выбрать наиболее близкий к полученному учащимся. Если полученный результат существенно отличается от всех предложенных вариантов, то это означает либо арифметическую ошибку, либо то, что надо выразить полученное значение в битах через байты или наоборот.

Пример (№ 2)

Каждый символ в Unicode закодирован двухбайтным словом. Оцените информационный объем следующего предложения в этой кодировке:

Без труда не вытащишь рыбку из пруда.

- 1) 37 бит 2) 592 бита 3) 37 байт 4) 592 байта

Решение

Длина фразы составляет примерно 40 символов. Следовательно, ее объем можно приблизительно оценить в $40 \times 2 = 80$ байт. Такого варианта ответа нет, попробуем перевести результат в биты: $80 \text{ байт} \times 8 = 640 \text{ бит}$. Наиболее близкое значение из предложенных — 592 бита. Заметим, что разница между 640 и 592 составляет всего $48/16 = 3$ символа в заданной кодировке и его можно считать несущественным по сравнению с длиной строки.

Ответ: 2.

Замечание: Подсчетом символов в строке можно убедиться, что их ровно 37 (включая точку и пробелы), поэтому оценка 592 бита = 74 байта, что соответствует ровно 37 символам в двухбайтовой кодировке, является точной.

При выполнении заданий, подобных № 4—10, следует пользоваться формулой алфавитного подхода к измерению количества информации $I = M \cdot \log_2 N$, где N — количество символов (мощность) алфавита, в котором записано сообщение, M — количество

символов в записи сообщения (длина сообщения), I — количество бит информации, содержащееся в сообщении. Если $\log_2 N$ не является целым числом, то I округляется в большую сторону.

Информационный объем сообщения, выраженный в битах, и минимальное количество двоичных разрядов, требуемое для записи сообщения в двоичном алфавите, совпадают.

Из приведенной формулы легко получить следующее следствие: с помощью n двоичных разрядов (бит) можно закодировать двоичным кодом все элементы множества мощностью 2^n (т.е. состоящего из 2^n элементов). Информационный объем одного символа алфавита, обозначающего элемент данного множества, будет равен n .

Пример (№ 11)

Метеорологическая станция ведет наблюдение за влажностью воздуха. Результатом одного измерения является целое число от 0 до 100%, которое записывается при помощи минимально возможного количества бит. Станция сделала 80 измерений. Определите информационный объем результатов наблюдений.

- 1) 80 бит 2) 70 байт 3) 80 байт 4) 560 байт

Решение

Способ 1

Воспользуемся приведенной выше формулой. Алфавитом в данном случае является множество целочисленных значений влажности от 0 до 100. Таких значений 101. Поэтому информационный объем результатов одного измерения $I = \log_2 101$. Это значение не будет целочисленным. Не вычисляя его, сразу найдем округленное в большую сторону целое значение. Заметим, что ближайшая к 101 целая степень двойки, большая 101, есть число $128 = 2^7$. Поэтому принимаем $I = \log_2 128 = 7$ бит. Учитывая, что станция сделала 80 измерений, общий информационный объем равен $80 \times 7 = 560$ бит = 70 байт.

Ответ: 2.

Способ 2

Воспользуемся следствием из формулы. Заметим, что $2^6 < 101 < 2^7$, поэтому минимально необходимое количество двоичных разрядов (бит) равно 7. Далее аналогично получаем, что общий информационный объем равен $80 \times 7 = 560$ бит = 70 байт.

Ответ: 2.

При выполнении заданий, связанных с понятием скорости передачи данных (№ 37—40), часто допускаются ошибки, связанные с неверным использованием размерности единиц измерения. Следует следить за размерностью исходных данных и размерностью, в которой требуется записать результат. Для успешного выполнения заданий такого типа нужно потренироваться в переводе Мбайт/мин в Кбайт/с и т.д.

Пример (№ 38)

Скорость передачи данных через ADSL-соединение равна 256 000 бит/с. Передача файла через данное соединение заняла 3 мин. Определите размер файла в килобайтах.

Решение

Размер файла = скорость \times время передачи. Выразим время в секундах, а скорость — в килобайтах в секунду.

Размер файла = $256\,000 / (8 \cdot 1024) \cdot 3 \cdot 60$ Кбайт.

Прежде чем выполнять действия, выделим в явном виде, там, где это очень просто, степени двойки.

Размер файла = $2^8 \cdot 1000 / (2^3 \cdot 2^{10}) \cdot 3 \cdot 15 \cdot 4 = 2^8 \cdot 125 \cdot 2^3 / (2^3 \cdot 2^{10}) \cdot 45 \cdot 2^2 = 2^{13} \cdot 125 \cdot 45 / 2^{13} = 125 \cdot 45 = 5625$ Кбайт.

Ответ: 5625.

Важное замечание:

Практически во всех заданиях можно избежать громоздких вычислений, упростив выражения, как это показано выше. Такая техника вычислений обязательно должна быть отработана в процессе подготовки к экзамену, поскольку она обеспечивает существенную экономию времени и минимум досадных арифметических ошибок.

Основные трудности при выполнении заданий на выполнение действий над числами в разных системах счисления (№ 24—27) порождаются недостаточным усвоением математического содержания понятия позиционной системы счисления. Для более глубокого понимания материала надо излагать алгоритмы перевода чисел из одной системы в другую с приведением доказательств.

Кроме того, рекомендуется побуждать учащихся к решению тренировочных заданий различными способами, с обязательным сравнением результатов. Необходимо выполнять проверку полученных результатов путем обратного перевода чисел или выполнения действий в другой системе счисления.

Для быстрого и правильного решения заданий ЕГЭ учащийся, помимо умения применять стандартные алгоритмы перевода чисел из одной системы счисления в другую, должен знать наизусть значения целых степеней числа 2 от 2^0 до 2^{10} , представление чисел от 0 до 16 в системах счисления с основаниями 2, 8, 10, 16, а также знать свойства систем счисления с основаниями вида $P = Q^n$ (в этом случае одной цифре в записи числа в системе с основанием P соответствует n цифр в системе с основанием Q).

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	54	128	256	512	1024

Основание	10	2	8	16
0	0	0	0	0
1	1	1	1	1
2	10	2	2	2
3	11	3	3	3
4	100	4	4	4
5	101	5	5	5
6	110	6	6	6
7	111	7	7	7
8	1000	10	8	8
9	1001	11	9	9
10	1010	12	A	A
11	1011	13	B	B
12	1100	14	C	C
13	1101	15	D	D
14	1110	16	E	E
15	1111	17	F	F
16	10000	20	10	10

Пример (№ 21)

Количество значащих нулей в двоичной записи десятичного числа 126 равно:

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

Решение

Способ 1

Преобразуем число 126 в двоичную систему с помощью известного алгоритма деления «уголком» с выделением остатков:

Алгоритмизация и программирование

Эта тема курса в экзаменационной работе 2007 г. была представлена наиболее подробно: в общей сложности 9 заданий базового, повышенного и высокого уровня сложности во всех трех разделах. Знания и умения, связанные с использованием основных алгоритмических конструкций, выявлялись как заданием на исполнение и анализ отдельных алгоритмов, записанных в виде блок-схемы, на алгоритмическом языке или на языках программирования, так и заданиями на составление алгоритмов для конкретного исполнителя (задание с кратким ответом) и анализ дерева игры.

Экзаменуемые в целом хорошо справились с заданиями на анализ и исполнение алгоритма, записанного в виде блок-схемы (средний процент выполнения около 80%) и на запись фрагмента алгоритма для исполнителя с фиксированным набором команд. Задание А7 на использование переменных также не вызвало затруднений. Задание повышенного уровня на алгоритмы работы с массивами правильно выполнили 64% экзаменуемых. Задание повышенного уровня сложности В6 вызвало затруднения (61% выполнения).

Два задания высокого уровня сложности с развернутым ответом показали следующий результат выполнения: 28% в среднем для задания на запись алгоритма на естественном языке или языке программирования и 36% в среднем для задания на анализ дерева игры.

В целом выполнение заданий этого раздела экзаменационной работы показало хорошее знание темы абитуриентами, что объясняется центральным положением данной темы в школьном курсе информатики и хорошо отработанным за долгие годы развития предмета содержанием обучения.

Приведем пример решения типичного задания на исполнение алгоритма, сформулированного на естественном языке.

Пример (№ 51)

Цепочки символов (строки) создаются по следующему правилу.

Первая строка состоит из одного символа — цифры «1».

Каждая из последующих цепочек создается такими действиями: в очередную строку дважды записывается цепочка цифр из предыдущей строки (одна за другой, подряд), а в конец приписы-

вается еще одно число — номер строки по порядку (на i -м шаге дописывается число « i »).

Вот первые 4 строки, созданные по этому правилу:

- 1) 1
- 2) 112
- 3) 1121123
- 4) 112112311211234

Какая цифра стоит в седьмой строке на 121-м месте (считая слева направо)?

Решение

Найдем длину седьмой строки. По условию длина каждой последующей строки увеличивается в 2 раза по сравнению с предыдущей плюс еще один символ — цифра, обозначающая порядковый номер самой строки.

Получается, что длина строк составит:

- 1) 1 элемент в строке;
- 2) $1 \times 2 + 1 = 3$ элемента в строке;
- 3) $3 \times 2 + 1 = 7$;
- 4) $7 \times 2 + 1 = 15$;
- 5) $15 \times 2 + 1 = 31$;
- 6) $31 \times 2 + 1 = 63$;
- 7) $63 \times 2 + 1 = 127$ элементов в строке.

Требуется найти 121-й элемент в строке длиной в 127 символов. Это означает, что нам нужен седьмой элемент с конца. Поскольку в конец строки на каждом шаге добавляется его номер (совпадающий с номером формируемой строки), то последние семь символов 7-й строки будут 1234567. Таким образом, седьмой символ с конца — единица.

Ответ: 1.

Для быстрого и успешного выполнения рассмотренного задания важно было не механически выполнить алгоритм, а понять закономерность, которую он выражает, и, воспользовавшись ей, найти решение.

Важное замечание:

Практически во всех заданиях на исполнение алгоритмов можно избежать большого объема рутинной работы, выявив закономерность, реализуемую алгоритмом.

Высоким уровнем сложности обладают задания, в которых требуется построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию (№ 52—54).

При выполнении таких заданий надо не только верно указать выигрывающего игрока и его стратегию, но и дать ей строгое обоснование, перебрав все варианты ходов обоих игроков, возможные при реализации одним из них своей выигрышной стратегии.

Пример

Два игрока играют в следующую игру.

Имеются три кучки камней, содержащих соответственно 2, 3, 4 камня. За один ход разрешается или удвоить количество камней в какой-нибудь кучке, или добавить по два камня в каждую из трех куч. Предполагается, что у каждого игрока имеется неограниченный запас камней.

Выигрывает тот игрок, после чьего хода в какой-нибудь кучке становится ≥ 15 камней или во всех трех кучках суммарно становится ≥ 25 камней.

Игроки ходят по очереди. Выясните, кто выигрывает при правильной игре, — первый или второй игрок.

Решение (развернутый ответ)

Для решения задачи составим таблицу (дерево развития игры при разных продолжениях); в колонке 0 показано начальное состояние игры (вершина дерева игры), в колонке 1 показаны 4 возможных состояния игры после 1-го хода 1-го игрока, в колонке 2 показано 16 возможных состояний игры после 1-го хода 2-го игрока, далее дерево игры не ведется, а проводится анализ уже рассчитанных состояний игры.

0	1	2	3	4
		8,3,4		
		4,6,4	проигрыш 1-го игрока (при любом продолжении)	
	4,3,4	4,3,8		
		6,5,6		
		4,6,4	проигрыш 1-го игрока (при любом продолжении)	
	2,6,4	2,12,4		
2,3,4		2,6,8		
		4,8,6		

2,3,8	проигрыш 1-го игрока (при ходе 2-го игрока 2,3,16)
8,5,6	выигрыш 1-го игрока (при ходе 16,5,6)
4,5,6	4,10,6 выигрыш 1-го игрока (при ходе 4,20,6)
	4,5,12 выигрыш 1-го игрока (при ходе 4,5,24)
6,7,8	выигрыш 1-го игрока (при ходе 6,7,16)

Если 1-й игрок сделает свой первый ход $2,3,4 \rightarrow 4,3,4$, то 2-й игрок при правильной игре сделает ход $4,3,4 \rightarrow 4,6,4$, что приводит к проигрышу 1-го игрока (т.к. из состояния $(4,6,4)$ 1-й игрок может своим ходом перевести игру в одно из четырех состояний — $(8,6,4)$, $(4,12,4)$, $(4,6,8)$, $(6,8,6)$, и для любого из этих состояний найдется ход 2-го игрока, дающий ему выигрыш, например, по критерию $S \geq 25$).

Если 1-й игрок сделает свой первый ход $2,3,4 \rightarrow 2,6,4$, то 2-й игрок при правильной игре сделает ход $2,6,4 \rightarrow 4,6,4$, что, как мы только что видели, приводит к выигрышу 2-го игрока.

Если 1-й игрок сделает свой первый ход $2,3,4 \rightarrow 2,3,8$, то его проигрыш очевиден, так как 2-й игрок, как указано в таблице, добьется выигрывающего состояния игры 2,3,16.

Наконец, если 1-й игрок сделает свой первый ход $2,3,4 \rightarrow 4,5,6$, то он выигрывает игру, т.к. на любой из четырех возможных ответов 2-го игрока (см. табл.) есть выигрывающий ход 1-го игрока.

Таким образом, при правильной игре выигрывает 1-й игрок (при этом его первый ход должен быть $2,3,4 \rightarrow 4,5,6$).

Приведем пример типовых критериев оценивания задания С3 (напомним, что оно оценивается из трех баллов).

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

Указания по оцениванию	Баллы
Правильное указание выигрывающего игрока и его ходов со строгим доказательством правильности (с помощью или без помощи дерева игры)	3
Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказательства ее правильности	2

Указания по оцениванию	Баллы
<p>При наличии в представленном решении одного из пунктов:</p> <p>1. Правильно указаны все варианты хода первого игрока и возможные ответы второго (в том числе и все выигрышные), но неверно определены дальнейшие действия и неправильно указан победитель.</p> <p>2. Правильно указан выигрывающий игрок, но описание выигрышной стратегии неполно и рассмотрены несколько (больше одного, но не все(!) вариантов хода первого игрока и частные случаи ответов второго игрока</p>	1
В представленном решении полностью отсутствует описание элементов выигрышной стратегии, и отсутствует анализ вариантов первого-второго ходов играющих (даже при наличии правильного указания выигрывающего игрока)	0
Максимальный балл	3

При решении заданий на исполнение алгоритма в среде формального исполнителя (**№ 65—71**) прежде всего требуется уяснить систему команд исполнителя алгоритма, т.е. как записывается каждая команда, что означают ее параметры (если они есть) и каков должен быть результат ее выполнения.

Пример

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 , вызывающая передвижение *Черепашки* на n шагов в направлении движения.

Направо m , вызывающая изменение направления движения на m градусов по часовой стрелке. $0 \leq m \leq 180$.

(Вместо n и m должны стоять целые числа).

Запись:

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

Какое число необходимо записать вместо n в следующем алгоритме:

Повтори 6 [Вперед 40 Направо n],

чтобы на экране появился правильный пятиугольник.

Решение

Сумма внутренних углов правильного пятиугольника вычисляется по формуле $(p-2) \times 180$, где $p=5$. Поэтому величина одного внутреннего угла будет равна $(5-2) \times 180/5 = 108^\circ$. А угол поворота Черепашки в вершине пятиугольника будет равен углу, смежному с внутренним углом, т.е. $180 - 108 = 72^\circ$.

Черепашка прочертит на экране 6 отрезков, но последний отрезок полностью совпадет с первым, так как после пятого выполнения цикла *Черепашка* полностью обернется вокруг своей оси ($72 \times 5 = 360^\circ$) и окажется в той же точке, что и изначально. Так что на экране появится правильный пятиугольник.

Ответ: 72.

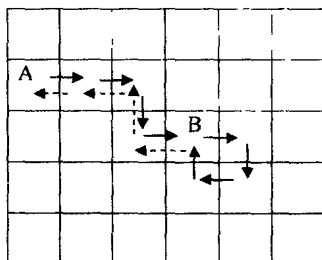
Пример (№ 71)

Исполнитель *Робот* действует на клетчатой доске, между соседними клетками которой могут стоять стены. *Робот* передвигается по клеткам доски и может выполнять команды 1 (вверх), 2 (вниз), 3 (вправо), 4 (влево), переходя на соседнюю клетку в направлении, указанном в скобках. Если в этом направлении между клетками стоит стена, то *Робот* разрушается. *Робот* успешно выполнил программу 33233241.

Какую последовательность из четырех команд должен выполнить *Робот*, чтобы вернуться в ту клетку, где он был перед началом выполнения программы, и не разрушится вне зависимости от того, какие стены стоят на поле?

Решение

Начертим траекторию, по которой двигался *Робот*. Обозначим буквой *A* — начальную точку его движения, *B* — конечную. Его путь из *A* в *B* показан непрерывными стрелками. Из рисунка видно, что возвращение из точки *B* в точку *A* можно осуществить по программе из четырех команд: влево-вверх-влево-влево, т.е. 4144. Обратный путь *Робота* показан пунктиром.

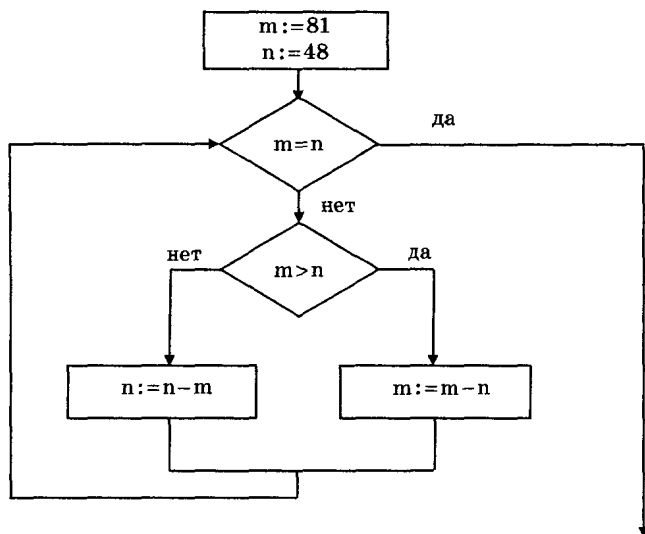


Ответ: 4144.

Для решения задач на исполнение алгоритма, записанного в виде блок-схемы или программы на алгоритмическом языке, нужно знать и уметь использовать основные алгоритмические конструкции: следование, ветвление, цикл. Для непосредственного исполнения алгоритма рекомендуется вести таблицу переменных, в которой отображается изменение их значений после каждого шага.

Пример (№ 57)

Определите значение переменной m после выполнения фрагмента алгоритма:



Примечание: знаком $:=$ обозначена операция присваивания.

- 1) 1 2) 2 3) 3 4) 33

Решение

Способ 1

Составим таблицу переменных, добавив в нее для удобства результаты вычисления логических выражений.

№ шага	Значение m	Значение n	$m = n$	$m > n$
0	81	48		
1	81	48	81=48 — нет (выполняем тело цикла)	

<i>N</i> шага	Значение <i>m</i>	Значение <i>n</i>	$m = n$	$m > n$
2	81	48		$81 > 48$ — да
3	33	48		
4	33	48	$33 = 48$ — нет (выполняем тело цикла)	
5	33	48		$33 > 48$ — нет
6	33	15		
7	33	15	$33 = 15$ — нет (выполняем тело цикла)	
8	33	15		$33 > 15$ — да
9	18	15	$18 = 15$ — нет (выполняем тело цикла)	
10	18	15		$18 > 15$ — да
11	3	15		
12	3	15	$3 = 15$ — нет (выполняем тело цикла)	
13	3	15		$3 > 15$ — нет
14	3	12		
15	3	12	$3 = 12$ — нет (выполняем тело цикла)	
16	3	12		$3 > 12$ — нет
17	3	9		
18	3	9	$3 = 9$ — нет (выполняем тело цикла)	
19	3	9		$3 > 9$ — нет
20	3	6		
21	3	6	$3 = 6$ — нет (выполняем тело цикла)	
22	3	6		$3 > 6$ — нет
23	3	3		
24	3	3	$3 = 3$ — да (выход из цикла и завершение алгоритма)	

Ответ: 3.

Способ 2

Внимательно проанализировав блок-схему, можно сделать вывод, что она реализует известный алгоритм Евклида нахождения наибольшего общего делителя двух чисел, который для 81 и 48 равен трем. ($81 = 3^4$, $48 = 3 \times 16$.)

Ответ: 3.

При выполнении заданий на выполнение алгоритмов, записанных на языках программирования, следует учесть, что приведенные в таблице тексты программ на разных языках эквивалентны, поэтому учащийся должен выбрать тот язык, который ему наиболее знаком, и далее работать только с ним, не обращая внимания на остальные столбцы таблицы.

Пример (№ 78)

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

Бейсик	Паскаль	Алгоритмический
$a = 2468$ $b = (a \text{ MOD } 1000) * 10$ $a = a \setminus 1000 + b$ \ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a := 2468;$ $b := (a \bmod 1000) * 10;$ $a := a \div 1000 + b;$ {div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}	$a := 2468$ $b := \text{mod}(a, 1000) * 10$ $a := \text{div}(a, 1000) + b$ div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

- 1) $a = 22$, $b = 20$
- 2) $a = 4682$, $b = 4680$
- 3) $a = 8246$, $b = 246$
- 4) $a = 470$, $b = 468$

Решение

Составим таблицу переменных:

№ шага	Значение a	Значение b
0	2468	не определено
1	2468	$468 * 10 = 4680$
2	$2 + 4680 = 4682$	4680

Ответ: 2.

В задачах на работу с одномерными и двумерными массивами надо определить закономерность заполнения результирующего массива, поскольку вычисление всех его значений «в лоб» займет значительное время. Однако для того чтобы понять эту закономерность, иногда требуется выполнить несколько итераций цикла «вручную».

Пример (№ 80)

Значения двух массивов $A[1..100]$ и $B[1..100]$ задаются с помощью следующего фрагмента программы:

Бейсик	Паскаль	Алгоритмический
FOR $n=1$ TO 100 $A(n)=n-10$ NEXT n FOR $n=1$ TO 100 $B(n)=A(n)*n$ NEXT n	for $n:=1$ to 100 do $A[n]:=n-10$; for $n:=1$ to 100 do $B[n]:=A[n]*n$	<u>нц</u> для n от 1 до 100 $A[n]=n-10$ <u>кц</u> <u>нц</u> для n от 1 до 100 $B[n]=A[n]*n$ <u>кц</u>

Сколько элементов массива B будут иметь положительные значения?

1) 10

2) 50

3) 90

4) 100

Решение

После завершения первого цикла массив A будет иметь вид: $-9, -8, -7, \dots, -1, 0, 1, 2, \dots, 90$

В нем будет 90 положительных значений. Умножение во втором цикле на положительные числа не изменит их знаки. Поэтому после второго цикла в массиве B также будет до положительных значений.

Ответ: 3.

Пример (№ 83)

Значения двумерного массива размера 7×7 задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический
FOR $n=1$ TO 7 FOR $k=1$ TO 7 $B(n, k)=k-n$ NEXT k NEXT n	for $n:=1$ to 7 do for $k:=1$ to 7 do $B[n, k]:=k-n$;	<u>нц</u> для n от 1 до 7 <u>нц</u> для k от 1 до 7 $B[n, k]=k-n$ <u>кц</u> <u>кц</u>

Сколько элементов массива будут иметь положительные значения?

1) 49

2) 28

3) 21

4) 7

Решение

Изобразим двумерный массив в виде таблицы. На диагонали таблицы, там, где $k=n$, будут находиться нули. Выше диагонали, там, где $n < k$, будут находиться положительные элементы, ниже диагонали — отрицательные.

$n \backslash k$	1	2	3	4	5	6	7
1	0	1		...			6
2	-1	0					
3		
4			
5					...		
6						0	1
7	-6			...		-1	0

Количество элементов над диагональю и под ней одинаково. И оно равно $(7 \times 7 - 7)/2 = 21$.

Ответ: 3.

Приведем пример решения задания на исполнение алгоритма, в котором используются функции работы над текстовыми строками. Для успешного выполнения таких заданий учащийся должен владеть понятием «тип данных» и уметь применить его на практике.

Пример (№ 62)

В приведенном ниже фрагменте алгоритма, записанном на алгоритмическом языке, переменные a, b, c имеют тип «строка», а переменные i, k — тип «целое». Используются следующие функции:

Длина (a) — возвращает количество символов в строке a . (Тип «целое».)

Извлечь (a, i) — возвращает i -й (слева) символ в строке a . (Тип «строка».)

Склеить (a, b) — возвращает строку, в которой записаны сначала все символы строки a , а затем все символы строки b . (Тип «строка».)

Значения строк записываются в одинарных кавычках. (Например, $a := \text{'дом'}$.)

Фрагмент алгоритма:

```

 $i := \text{Длина}(a)$ 
 $k := 1$ 
 $b := \text{'П'}$ 
пока  $i > 0$ 
нц
 $c := \text{Извлечь}(a, i)$ 
 $b := \text{Склеить}(b, c)$ 
 $i := i - k$ 
кц

```

Какое значение будет у переменной b после выполнения вышеприведенного фрагмента алгоритма, если значение переменной a было 'РОЗА'?

- 1) 'ПАЗ' 2) 'ПАЗОР' 3) 'ПОЗА' 4) 'ПРОЗА'

Решение

В данном случае для решения задачи достаточно знания обычного алгоритмического языка и описания функций, приведенного в условии. Выполним программу по шагам, занося значения переменных в таблицу:

Выполняемый оператор	Значение a	Значение b	Значение c	Значение i	Значение k
	'РОЗА'	не определено	не определено	не определено	не определено
$i := \text{Длина}(a)$	'РОЗА'	не определено	не определено	4	не определено
$k := 1$	'РОЗА'	не определено	не определено	4	1
$b := \text{'П'}$	'РОЗА'	'П'	не определено	4	1
$c := \text{Извлечь}(a, i)$	'РОЗА'	'П'	'А'	4	1
$b := \text{Склеить}(b, c)$	'РОЗА'	'ПА'	'А'	4	1

Выполняе- мый опера- тор	Значе- ние a	Значение b	Значение c	Значе- ние i	Значение k
$i := i - k$	‘РОЗА’	‘ПА’	‘А’	3	1
$c :=$ Извлечь (a, i)	‘РОЗА’	‘ПА’	‘З’	3	1
$b :=$ Склеить (b, c)	‘РОЗА’	‘ПАЗ’	‘З’	3	1
$i := i - k$	‘РОЗА’	‘ПАЗ’	‘З’	2	1
$c :=$ Извлечь (a, i)	‘РОЗА’	‘ПАЗ’	‘О’	2	1
$b :=$ Склеить (b, c)	‘РОЗА’	‘ПАЗО’	‘О’	2	1
$i := i - k$	‘РОЗА’	‘ПАЗО’	‘О’	1	1
$c :=$ Извлечь (a, i)	‘РОЗА’	‘ПАЗО’	‘Р’	1	1
$b :=$ Склеить (b, c)	‘РОЗА’	‘ПАЗОР’	‘Р’	1	1
$i := i - k$	‘РОЗА’	‘ПАЗОР’	‘Р’	0	1

Ответ: 2.

К теме «Алгоритмизация и программирование» относятся также задания (№ 86—91) на проверку умения написать короткую (10—15 строк) простую программу обработки массива на языке программирования или записать алгоритм на естественном языке.

Подчеркнем, что в данном задании от ученика не обязательно требуется писать программу на языке программирования. Для получения высшего балла за задание достаточно сформулировать корректный алгоритм на естественном языке.

Для решения этой группы заданий полезно усвоить следующие элементарные алгоритмы:

1) Поиск минимального и максимального элементов в массиве с определением их номеров.

2) Поиск в массиве элемента, удовлетворяющего заданному условию, с определением его номера.

3) Подсчет числа элементов массива, удовлетворяющих заданному условию.

4) Вычисление суммы элементов числового массива.

5) Вычисление суммы элементов числового массива, удовлетворяющих заданному условию.

6) Поиск в массиве подпоследовательности убывающих (возрастающих) элементов.

После того как алгоритм начерно написан, рекомендуется его протестировать на небольших (4—5 элементов) массивах исходных данных. Желательно проверить корректность работы алгоритма в следующих ситуациях:

1) элементы массива различны и не упорядочены;

2) элементы массива различны и упорядочены по возрастанию;

3) элементы массива различны и упорядочены по убыванию;

4) элементы массива равны между собой;

5) иные «экстремальные» случаи.

Типичными ошибками при составлении алгоритмов работы с массивами как на естественном языке, так и на языках программирования является отсутствие инициализации индексных переменных, переменных-счетчиков и переменных-сумматоров; выход за пределы границ массива.

Распространенной ошибкой при составлении алгоритмов работы с массивом на естественном языке является неоправданное укрупнение шагов алгоритма.

Так, например, решение задачи нахождения разности максимального и минимального элементов массива в виде «найдем максимальный и минимальный элементы, затем вычислим их разность», из-за его существенной неполноты нельзя признать удовлетворительным и положительно оценить.

Приведем пример решения типичного задания C2 с критериями оценивания ответов.

Пример (№ 86)

Опишите на русском языке или одном из языков программирования алгоритм подсчета суммы произведений последовательных пар элементов в целочисленном массиве из 30 элементов.

(Это означает, что надо сосчитать сумму произведений первого и второго, третьего и четвертого, пятого и шестого элементов, и т.д.)

Ответ:

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Введем целочисленную переменную SumProd, в которую будем заносить сумму произведений последовательных пар элементов в просмотренной части массива, и присвоим ей начальное значение 0. В цикле до конца массива: считаем произведение элементов очередной пары и прибавляем его к SumProd. По окончании цикла переменная SumProd содержит сумму произведений последовательных пар элементов массива.</p> <p>Пример правильной и эффективной программы (на основе алгоритма, использующего однократный проход по массиву):</p>	
На языке Паскаль	На языке Бейсик
<pre>const N=30; var a:array[1..N] of integer; SumProd, i: integer; begin SumProd:=0; for i:=1 to N div 2 do begin SumProd:=SumProd+a[2*i-1]*a[2*i]; end; writeln(SumProd); end</pre>	<pre>N=30 DIM i, SumProd, a(N) AS INTEGER SumProd=0 FOR i=1 TO N/2 SumProd=SumProd+a(2*i-1)*a(2*i) NEXT i PRINT SumProd END</pre>

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение (в том числе и алгоритм, требующий двукратного прохода по массиву или создания массива произведений пар).</p> <p>Возможно использование числа 30 вместо константы.</p> <p>Возможно наличие отдельных синтаксических ошибок (пропущенные «;», неверная запись оператора присваивания и т.п.), не искажающих замысла автора программы</p>	2
<p>Имеется не более двух ошибок из числа следующих:</p> <p>1) Не задано первое значение SumProd.</p> <p>2) Не указано или неверно указано условие завершения цикла.</p>	1

Указания по оцениванию	Баллы
3) Индексная переменная в цикле не меняется. 4) Неверно расставлены операторные скобки	1
<i>Ошибок, перечисленных выше, больше двух, или алгоритм сформулирован неверно (в частности, если хотя бы один из элементов массива умножается более одного раза)</i>	0
<i>Максимальный балл</i>	2

Основы логики

По данной теме в экзаменационной работе 2007 г. содержалось 5 заданий: 3 с выбором ответа и 2 с кратким ответом. Одно задание базового, три повышенного и одно — высокого уровня сложности. Учащиеся хорошо справились с заданием A11 повышенного уровня на проверку умения строить таблицы истинности и логические схемы (в среднем 78% выполнения), с заданием базового уровня на преобразование логических выражений (79% правильных ответов). Хуже было выполнено задание повышенного уровня на проверку знания основных понятий и законов математической логики — 57%. Тем не менее результаты экзамена на протяжении всех трех лет его проведения показывают стабильный рост результатов, что говорит о безусловном влиянии ЕГЭ на содержание школьного образования.

В 2005 г. в ЕГЭ впервые предлагалась текстовая логическая задача как задание с кратким ответом. В тот год в целом учащиеся неплохо справились с заданием, учитывая его повышенный уровень — 48% выполнения в среднем, а в 2006 г. 57% выполнения, в 2007 г. — 64%.

В целом в 2007 г. продолжился проявившийся еще на экзамене 2005 г. рост результатов выполнения по теме «Основы логики», что, по-видимому, связано с тем, какое большое внимание было уделено этому разделу при разборе результатов ЕГЭ предыдущих лет. На сегодняшний день можно констатировать, что данная тема перестала быть «сверхтрудной», результаты ее выполнения экзаменуемыми в общем соответствуют результатам по первым двум темам.

Для успешного выполнения заданий ЕГЭ по основам логики учащиеся должны твердо усвоить символику и определения (таблицы истинности) трех основных логических операций (инверсия, конъюнкция, дизъюнкция), а также импликации. Кроме того, необходимо знать и уметь применять при работе с логическими выражениями основные законы логики. Приведем их перечень:

Название закона	Формулировка
Переместительный закон	$A \vee B = B \vee A$ $A \wedge B = B \wedge A$
Сочетательный закон	$(A \vee B) \vee C = A \vee (B \vee C)$ $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
Распределительный закон	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
Закон непротиворечия. Этот закон выражает тот факт, что высказывание не может быть одновременно истинным и ложным	$A \wedge \neg A = 0$
Закон исключенного третьего. Этот закон означает, что либо высказывание, либо его отрицание должно быть истинным	$A \vee \neg A = 1$
Закон двойного отрицания	$\neg(\neg A) = A$
Законы де Моргана	$\neg(A \vee B) = \neg A \wedge \neg B$ $\neg(A \wedge B) = \neg A \vee \neg B$

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

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

Кроме того, желательно знать следующие свойства конъюнкции, дизъюнкции и импликации:

$$A \vee 0 = A;$$

$$A \wedge 0 = 0;$$

$$A \vee 1 = 1;$$

$$A \wedge 1 = A;$$

$$0 \rightarrow A = 1.$$

Рассмотрим на примерах способы решения типичных заданий.

Пример (№ 98)

Какое логическое выражение равносильно выражению

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

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

Решение

Применим отрицание к выражению в скобках в соответствии с законом де Моргана:

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

Ответ: 2.

Пример (№ 115)

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

Дан фрагмент таблицы истинности выражения F :

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

Какое выражение соответствует F ?

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

Решение

Последовательно подставим первую строку таблицы истинности во все варианты ответов:

1) $\neg 0 \vee 1 \vee \neg 0 = 1$, а по условию F для этого набора значений равно 0. Первый ответ не подходит.

2) $0 \wedge \neg 1 \wedge \neg 0 = 0$, по условию $F = 0$. Второй ответ пока подходит.

3) $\neg 0 \wedge 1 \wedge 0 = 0$, по условию $F = 0$. Третий ответ пока подходит.

4) $0 \vee \neg 1 \vee 0 = 0$, по условию $F = 0$. Четвертый ответ пока подходит.

Отбросив первый вариант ответа, подставим теперь вторую строку во все оставшиеся:

2) $0 \wedge \neg 1 \wedge \neg 1 = 0$, по условию $F = 1$. Второй ответ отпадает.

3) $\neg 0 \wedge 1 \wedge 1 = 1$, по условию $F = 1$. Третий ответ пока подходит.

4) $0 \vee \neg 1 \vee 1 = 1$, по условию $F = 1$. Четвертый ответ пока подходит.

Подставим теперь третью строку в оставшиеся два варианта ответов:

3) $\neg 1 \wedge 1 \wedge 0 = 0$, по условию $F = 0$. Третий ответ подходит для всех строк.

4) $1 \vee \neg 1 \vee 0 = 1$, по условию $F = 0$. Четвертый ответ не подходит.

Ответ: 3.

Пример (№ 92)

Для какого числа X истинно высказывание:

$$\neg ((X > 3) \rightarrow (X > 4))$$

1) 1

2) 2

3) 3

4) 4

Решение

Способ 1

Поочередно подставляем значения X в высказывание:

1) $\neg ((1 > 3) \rightarrow (1 > 4)) = \neg (\text{ложь} \rightarrow \text{ложь}) = \neg (\text{истина}) = \text{ложь}$;

2) $\neg ((2 > 3) \rightarrow (2 > 4)) = \neg (\text{ложь} \rightarrow \text{ложь}) = \neg (\text{истина}) = \text{ложь}$;

3) $\neg ((3 > 3) \rightarrow (3 > 4)) = \neg (\text{ложь} \rightarrow \text{ложь}) = \neg (\text{истина}) = \text{ложь}$;

3) $\neg ((4 > 3) \rightarrow (4 > 4)) = \neg (\text{истина} \rightarrow \text{ложь}) = \neg (\text{ложь}) = \text{истина}$.

Ответ: 4.

Способ 2

По условию $\neg ((X > 3) \rightarrow (X > 4)) = \text{истина}$, поэтому $((X > 3) \rightarrow (X > 4)) = \text{ложь}$.

Из определения импликации следует, что высказывание $(X > 3)$ истинно, а $(X > 4)$ — ложно. Этим условиям из целых чисел удовлетворяет только 4.

Ответ: 4.

Аналогичные рассуждения, основанные на свойствах импликации, применяются при решении № 93—77.

Рассмотрим приемы решения текстовой логической задачи.

Пример

Три свидетеля дорожного происшествия сообщили сведения о скрывшемся нарушителе. Боб утверждает, что тот был на красном «Рено», Джон сказал, что нарушитель уехал на синей «Тойоте», а Сэм показал, что машина была точно не красная и, по всей видимости, это был «Форд». Когда удалось отыскать машину, выяснилось, что каждый из свидетелей точно определил только один из параметров автомобиля, а в другом ошибся. Какая и какого цвета была машина у нарушителя?

Ответ запишите в виде двух слов, разделенных пробелом: **МАРКА ЦВЕТ**. Например: **ЖИГУЛИ БЕЛЫЙ**.

Решение

Способ 1

Обозначим высказывания:

A = «машина красного цвета»;

B = «машина была «Рено»;

C = «машина синего цвета»;

D = «машина была «Тойота»;

E = «машина была «Форд».

Согласно условию:

из показаний Боба следует, что $A \vee B$ истинно;

из показаний Джона следует, что $C \vee D$ истинно;

из показаний Сэма следует, что $\neg A \vee E$ истинно.

Следовательно, истинна и конъюнкция $(A \vee B) \wedge (C \vee D) \wedge (\neg A \vee E) = 1$.

Раскрывая скобки, получаем:

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

Из полученных восьми слагаемых семь (согласно условию) являются ложными, остается единственное истинное слагаемое (подчеркнуто):

$$B \wedge C \wedge \neg A = 1.$$

Значит, нарушитель скрылся на автомобиле «Рено» синего цвета.

Ответ: РЕНО СИНИЙ.

Способ 2

Решим задачу методом рассуждений.

Предположим, что Боб правильно сообщил цвет, но ошибся в марке. Следовательно, машина красная, и не «Рено». Тогда получается, что Джон ошибся в цвете, но верно сообщил марку — «Тойота». Итак, предварительный вывод — красная «Тойота». Но при этом получается, что вопреки условиям задачи Сэм ошибся и в цвете, и в марке. Мы пришли к противоречию, значит, исходное предположение было неверным. Отсюда мы заключаем, что Боб верно указал марку — «Рено», но ошибся в цвете. Итак, машина «Рено», но не красного цвета. Учитывая, что машина точно не «Тойота», из показаний Джона вытекает, что машина была синей. При этом также выполняется условие для показаний Сэма.

Ответ: РЕНО СИНИЙ.

Моделирование

По теме «Моделирование» на экзамене в 2007 г. было только одно задание базового уровня с выбором ответа, которое учащиеся очень хорошо выполнили: средний процент выполнения выше 80%.

Рассмотрим примеры решения типичных заданий по этой теме.

Пример (№ 119)

Таблица стоимости перевозок устроена следующим образом: числа, стоящие на пересечениях строк и столбцов таблиц, означают стоимость проезда между соответствующими соседними станциями. Если пересечение строки и столбца пусто, то станции не являются соседними.

Укажите таблицу, для которой выполняется условие: «Минимальная стоимость проезда из A в B не больше 6».

Стоимость проезда по маршруту складывается из стоимостей проезда между соответствующими соседними станциями.

1)

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

2)

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

3)

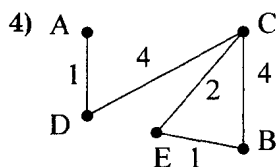
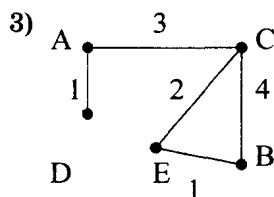
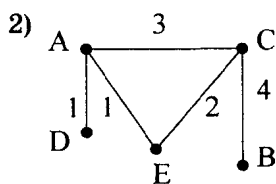
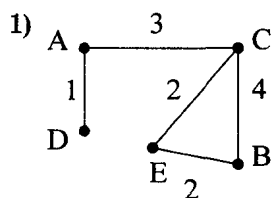
	A	B	C	D	E
A			3	1	
B			4		1
C	3	4			2
D	1				
E		1	2		

4)

	A	B	C	D	E
A				1	
B			4		1
C		4		4	2
D	1		4		4
E		1	2		

Решение

Построим схемы, соответствующие каждой таблице:



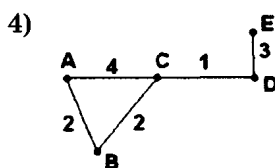
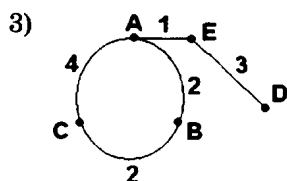
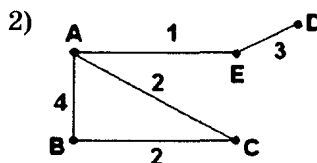
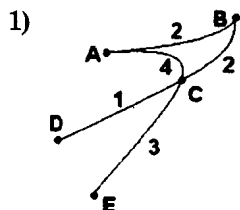
Видно, что минимальная стоимость проезда из A в B достигается на схеме 3 на маршруте $ACEB$, и она равна 6, т.е. условие задания выполнено.

Ответ: 3.

Пример (№ 120)

В таблице приведена стоимость перевозок между соседними железнодорожными станциями. Укажите схему, соответствующую таблице.

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



Решение

Построим таблицу для каждой схемы:

1)

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

2)

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

3)

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

4)

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

Построенная для третьей схемы таблица совпадает с таблицей из условия.

Ответ: 3.

РАЗДЕЛ 2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

Общее замечание

Целью ЕГЭ не является проверка знаний и умений учащихся применительно к конкретным программным продуктам, операционным системам или технологиям определенных фирм-производителей компьютерного оборудования и программного обеспечения. В связи с этим контрольные измерительные материалы по данной теме ориентированы на проверку знаний об общих, инвариантных закономерностях тех или иных информационных технологий.

Программные средства информационных и коммуникационных технологий

Задания этой темы имеют традиционно высокий процент выполнения, что, по-видимому, объясняется большой практической востребованностью данной компоненты курса у школьников. Можно предположить, что знания и умения, получаемые при изучении данной темы, у подавляющего большинства учащихся имеют закрепление в ходе учебной и повседневной деятельности.

Для успешного выполнения заданий надо знать классификацию программного обеспечения, свойства и функциональные возможности основных видов программного обеспечения, структуру файловой системы, включая правила именования каталогов и файлов. Следует понимать разницу между именем файла, путем к нему, полным (абсолютным) и относительным именем.

Разберем типичное задание по этой теме:

Пример (№ 124)

Перемещаясь из одного каталога в другой, пользователь последовательно посетил каталоги **DOC**, **USER**, **SCHOOL**, **A:**, **LETTER**, **INBOX**. Каково полное имя каталога, в котором оказался пользователь?

Примечание: при каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше.

- 1) INBOX
- 2) A:\ LETTER\INBOX
- 3) A:\SCHOOL\USER\DOC
- 4) LETTER\INBOX

Решение

A:\ — имя корневого каталога, поэтому, последовательно двигаясь из него, пользователь попал в каталог A:\ LETTER\INBOX.

Ответ: 2.

Обработка графической информации

Основные проверяемые элементы этой темы — знание принципов векторной и растровой графики, в том числе способов компьютерного представления векторных и растровых изображений, умение оперировать с понятиями «глубина цвета», «пространственное и цветовое разрешение изображений и графических устройств», «кодировка цвета», «графический объект», «графический примитив», «пиксель».

Рассмотрим типичные задания данной темы.

Пример

Для хранения растрового изображения размером 128×128 пикселей отвели 4 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

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

Решение

Подсчитаем количество пикселей в изображении:

$$128 \times 128 = 2^7 \times 2^7 = 2^{14}.$$

Вычислим объем памяти в битах: $4 \text{ Кб} = 4 \times 2^{10} \text{ байт} = 2^2 \times 2^{10} \times 2^3 = 2^{15} \text{ бит}.$

Таким образом, на один пиксель изображения приходится $2^{15}/2^{14} = 2 \text{ бита}.$

Как известно, двумя двоичными разрядами можно закодировать четыре разных состояния объекта, в данном случае четыре цвета пикселя.

Ответ: 4.

Еще раз обращаем внимание на необходимость использовать свойства степенной функции вместо вычисления «в лоб».

Приведем пример решения *обратной* задачи.

Пример

Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 32×32 пикселя, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно.

1) 1

2) 2

3) 64

4) 1024

Решение

Исходя из количества цветов в палитре определим минимальное количество двоичных разрядов, необходимое для хранения одного пикселя. Для представления 256 различных состояний требуется $\log_2 256 = 8$ двоичных разрядов, т.е. 1 байт.

Поэтому для представления изображения размером 32×32 пикселя потребуется $32 \times 32 = 2^5 \times 2^5 = 2^{10}$ байт информации, т.е. 1 Кб.

Ответ: 1.

Обработка информации в электронных таблицах

Для решения заданий этой темы надо знать правила адресации ячеек в электронной таблице, знать различие между абсолютной и относительной адресацией и уметь использовать его на практике.

В электронных таблицах принято следующее правило: обычные адреса ячеек в формулах являются относительными. Это означает, что при копировании ячейки, содержащей формулу, в ячейку, отстоящую от исходной на некоторое число столбцов и строк, адреса ячеек в формуле изменяются на такое же число столбцов и строк. Пусть, например, ячейка B2 содержит формулу $=C2+1$. При копировании ячейки формула изменится следующим образом:

	A	B	C	D
1	=B1+1		=D1+1	
2		=C2+1		=E2+1
3	=B3+1			
4			=D4+1	

Чтобы адрес при копировании не менялся, он должен быть абсолютным. В абсолютном адресе перед обозначениями строки и столбца ставится знак \$. Если знак \$ стоит только перед именем столбца, то при копировании будет сохраняться имя столбца, если перед номером строки — номер строки.

Пусть ячейка B2 содержит формулу =\$C\$2+\$C3+C\$4. При копировании ячейки формула изменится следующим образом:

	A	B	C	D
1	=C\$2+\$C2+B\$4		=C\$2+\$C2+D\$4	
2		=C\$2+\$C3+C\$4		=C\$2+\$C3+E\$4
3	=C\$2+\$C4+B\$4			
4			=C\$2+\$C5+D\$4	

Пример

В ячейке A1 электронной таблицы записана формула =\$A2+C1. Какой вид приобретет формула после того, как ячейку A1 скопируют в ячейку B1?

- 1) =\$B2+D1
- 2) =\$A2+D1
- 3) =\$A2+D2
- 4) =\$B2+C1

Решение

Адрес B1 получен из адреса A1 сдвигом на 1 вправо. Также изменятся все относительные адреса столбцов в адресах формулы, а именно C1 преобразуется в D1. Адреса строк не изменятся, так как формула копируется в пределах одной строки (первой). Адрес \$A2 не изменится, так как здесь адрес столбца абсолютный, и формула приобретет вид =\$A2+D1.

Ответ: 2.

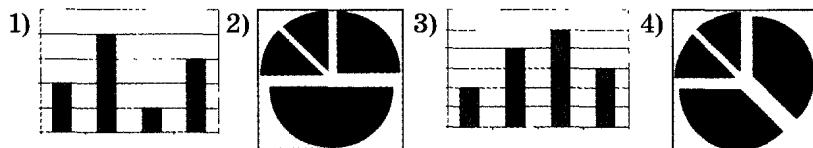
Отдельную группу задания по этой теме образуют задачи на представление числовых данных в виде диаграмм. Для решения этих задач нужно уметь строить диаграммы столбчатого и кругового типа, выражающие как абсолютные величины, так и относительные зависимости между исходными данными. Также необходимо уметь анализировать и сопоставлять диаграммы.

Пример (№ 145)

Дан фрагмент электронной таблицы:

	А	В
1	$=B1+1$	1
2	$=A1+2$	2
3	$=B2-1$	
4	$=A3$	

После выполнения вычислений была построена диаграмма по значениям диапазона ячеек **A1:A4**. Укажите получившуюся диаграмму.



Решение

Вычислив значения указанного диапазона, получим:

	А	В
1	2	1
2	4	2
3	1	
4	1	

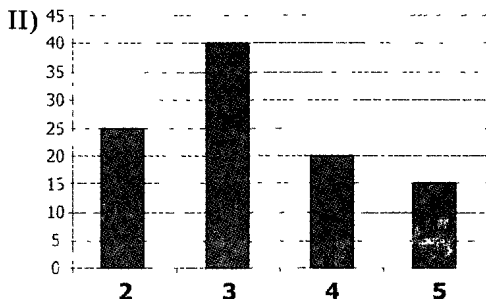
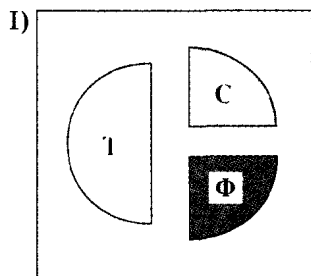
Соотношение величин **2 : 4 : 1 : 1** соответствует круговой диаграмме под номером 2.

Ответ: 2.

Приведем пример задания, в котором требуется интерпретировать и сопоставлять данные диаграмм различных типов:

Пример (№ 151)

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



Имеются четыре утверждения:

А) Среди слесарей найдется хотя бы один третьего разряда.

Б) Среди токарей найдется хотя бы один второго разряда.

В) Все токари могут иметь четвертый разряд.

Г) Все фрезеровщики могут иметь третий разряд.

Какое из этих утверждений следует из анализа обеих диаграмм?

1) А

2) Б

3) В

4) Г

Решение

По правой диаграмме найдем общее количество рабочих $25 + 40 + 20 + 15 = 100$ чел.

Из левой диаграммы следует, что токарей половина от общего количества рабочих, т.е. 50 чел., слесарей и фрезеровщиков — по 25 чел.

Проверим утверждение А. Оно не следует из анализа диаграммы, так как слесарей меньше, чем суммарное количество работников второго, четвертого и пятого разрядов, следовательно, возможна ситуация, когда никто из слесарей не имеет третий разряд.

Проверим утверждение Б. Оно не следует из анализа диаграммы по той же причине, что и А.

Утверждение В тоже ложно, т.к. токарей больше, чем рабочих четвертого разряда.

Утверждение Г — истинно, так как фрезеровщиков насчитывается 25 человек, следовательно, все они могут входить в число 40 рабочих, имеющих третий разряд.

Ответ: 4.

Хранение, поиск и сортировка информации в базах данных

Успешное решение заданий по этой теме обусловлено знанием принципов организации табличных (реляционных) баз данных, владением понятиями «таблица», «запись таблицы», «поле записи», «значение поля». Важно понимать, что каждая строка таблицы базы данных (БД) представляет собой целостный объект, объединяющий элементы с разными, как правило, типами данных — запись, с которым можно производить действия путем запросов к БД. Нужно отличать общие для всех записей таблицы наименования полей от их значений в отдельных записях. Основными операциями, встречающимися в заданиях этой темы, являются отбор (поиск) записей по некоторым условиям и их сортировка.

Пример (№ 158)

Далее в табличной форме представлен фрагмент базы данных о результатах тестирования учащихся (используется стобалльная шкала):

Фамилия	Пол	Математика	Русский язык	Химия	Информатика	Биология
Аганиян	ж	82	56	46	32	70
Воронин	м	43	62	45	74	23
Григорчук	м	54	74	68	75	83
Роднина	ж	71	63	56	82	79
Сергеенко	ж	33	25	74	38	46
Черепанова	ж	18	92	83	28	61

Сколько записей в данном фрагменте удовлетворяют условию «Пол=’м’ ИЛИ Химия>Биология»?

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

Решение

Решая данную задачу, надо последовательно применять условие к каждой строке таблицы. Условию удовлетворяют учащиеся: Воронин, Григорчук, Сергеенко и Черепанова. У Аганиян и Родниной не соответствует условию ни пол, ни соотношение баллов по химии и биологии.

Ответ: 4.

Телекоммуникационные технологии

В этой теме можно выделить две наиболее актуальные компоненты — адресация и поиск информации в Интернете.

Основные ошибки при выполнении заданий первой компоненты связаны с недостаточно глубоким пониманием частью учащихся правил адресации с использованием обычных унифицированных указателей ресурсов (URL — Uniform Resource Locator), иначе говоря, URL-адресов.

Пример (№ 163)

Доступ к файлу uk.net, находящемуся на сервере org.de, осуществляется по протоколу ftp. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

А	Б	В	Г	Д	Е	Ж
org	uk	.de	.net	://	ftp	/

Решение: Адрес ресурса начинается с названия протокола, в данном случае — это ftp (буква Е). Имя протокола должно отделяться от имени сервера двоеточием и двумя наклонными чертами (Д). Имя сервера — org.de кодируется буквами А и В. После имени сервера следует наклонная черта (Ж), отделяющая его от имени файла (Б,Г). Итак, полный путь к файлу: ftp://org.de/uk.net, соответствующая последовательность букв: ЕДАВЖБГ.

Ответ: ЕДАВЖБГ.

Ошибки учащихся при выполнении заданий второй компоненты связаны с неправильным пониманием роли логических операций в конструировании поисковых запросов. При разборе этой темы целесообразно провести аналогию между сетью Интернет и базой данных и на упрощенном примере показать, что механизм влияния логических связей на результаты запроса для маленькой базы данных и для глобальной сети Интернет действует сходным образом.

Пример (№ 165)

В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке **возрастания** коли-

чества страниц, которые найдет поисковый сервер по каждому запросу.

Для обозначения логической операции “ИЛИ” в запросе используется символ $|$, а для логической операции “И” — $\&$.

А	волейбол баскетбол подача
Б	волейбол баскетбол подача блок
В	волейбол баскетбол
Г	волейбол & баскетбол & подача

Решение

Способ 1

Связка **И** между двумя словами в поисковом запросе означает, что требуется найти web-страницы, содержащие одновременно и первое, и второе слово. Связка **ИЛИ** — что ищутся страницы, включающие хотя бы одно из указанных слов. Поэтому больше всего страниц будет найдено по запросу Б, так как в искомое множество страниц попадут все страницы, каждая из которых содержит хотя бы одно (любое) слово из поискового запроса.

Меньше всего страниц будет найдено по запросу Г, поскольку он требует присутствия на искомой странице *всех* трех слов одновременно.

По запросу А будет найдено больше страниц, чем по запросу В, из-за этого в результаты запроса А войдут страницы, содержащие слово «подача», которые не попадут в результаты выполнения запроса В, если в них не будет слов «волейбол» и «баскетбол». Так, например, если на странице есть словосочетание «подача в теннисе», но нет ни слова про волейбол и баскетбол, то она будет найдена по запросам А и Б, но не будет найдена по запросам В и Г.

Ответ: ГВАБ.

Способ 2

Рассмотрим множества web-страниц, содержащие каждое из искомых слов. Запросу $X\&Y$ будет соответствовать пересечение множеств X и Y , а запросу $X|Y$ — их объединение. Воспользуемся графическим представлением действий над множествами. Множество страниц, содержащих некоторое слово, будем обозначать эллипсом. Множество, получившееся в результате запроса, будем закрашивать серым цветом.

Диаграмма для запроса А
будет выглядеть следующим
образом:

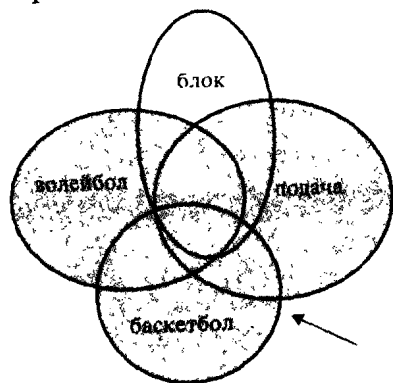


Диаграмма для запроса Б:

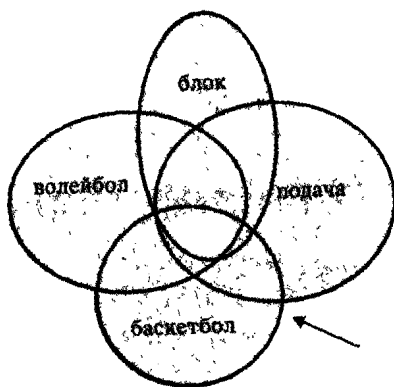


Диаграмма для запроса В:

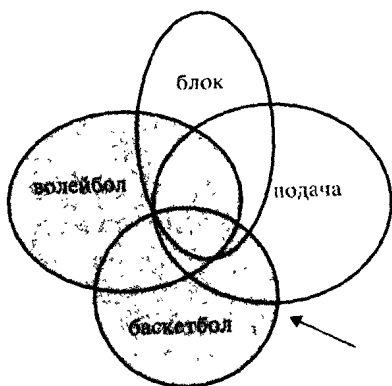
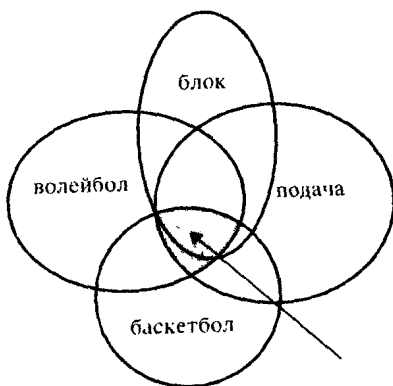


Диаграмма для запроса Г:



Получается, что результаты запроса возрастают в порядке
ГВАБ.

Ответ: ГВАБ.

Технологии программирования

Знания учащихся по технологии программирования в 2007 г. проверялись при помощи двух заданий с развернутым ответом. Одно задание повышенного уровня сложности (C1) предполагало поиск и устранение ошибок в фрагменте программы, другое (C4) предполагало самостоятельное написание программы для

решения оригинальной задачи (высокий уровень сложности). Около 20% от всех участников экзамена решили задание С1 полностью, набрав 3 балла, а 18% участников либо не приступали к ее решению, либо не смогли ничего сделать. Выпускники 2007 года при выполнении задания на поиск ошибок набрали в среднем 39% от максимально возможного балла за это задание (при 48% в 2006 г.), в то время как аналогичный средний процент выполнения задачи С4 на самостоятельное программирование — 10% (при 12% в 2006 г.). Полностью же решили задание С4, набрав 4 балла, только 3% от всех участников, более 75% не приступали к ее решению и только 12% выпускников смогли выполнить какую-то часть задания, получив от 1-го до 3-х баллов.

Учителя школ часто заявляют о недостатке времени на изучение программирования в школе. Вузы во многом интересуют только результаты выполнения заданий по этому разделу. Единый государственный экзамен еще раз выявил разрыв в требованиях школ и вузов к результатам обучения в средней школе.

Рассмотрим более подробно методику выполнения задания С1.

Прежде всего, следует четко уяснить задачу, которую должна решать программа. Без этого поиск ошибок в решении не имеет смысла. После того как выяснена цель программы, можно приступать к анализу ее текста. Не следует тратить время на анализ и сравнение всех трех вариантов программы на разных языках. Эти варианты алгоритмически идентичны, авторами задания в них внесены одни и те же логические ошибки. Три варианта программы (на Бейсике, Паскале и Си) приводится только для того, чтобы учащийся мог выбрать наиболее знакомый ему язык программирования и решать задачу на этом языке. Далее следует разобраться, насколько программа соответствует намеченной цели, т.е. делает ли она для всех наборов входных данных то, что от нее требуется, и не выполняет ли она при этом каких-либо лишних действий. Не стоит пытаться искать синтаксические ошибки, т.е. ошибки в написании служебных слов языка программирования или в расстановке разделительных знаков, а также ошибки выполнения, которые могут возникнуть из-за конкретной машинной реализации языка (переполнение регистров при умножении и т.д.). Речь идет только об ошибках в логике алгоритма. Если не удается сразу их найти, то можно попробовать выполнить про-

грамму для различных исходных данных на черновике, при необходимости предварительно составив блок-схему программы. Создание блок-схемы позволяет абстрагироваться от конкретного языка программирования и сосредоточиться на анализе алгоритма.

Чтобы продемонстрировать обнаруженную ошибку, нужно указать такие значения входных данных, при которых из-за этой ошибки программа дает неверный результат. После того как все ошибки найдены, можно переходить к их исправлению.

После исправления текста следует убедиться, что программа стала правильно работать на тех исходных данных, для которых она неправильно работала, и при этом не утратила способности правильно работать на тех данных, с которыми она верно работала и до внесения исправлений.

При подборе тестовых исходных данных следует руководствоваться следующими правилами.

Обязательно следует рассматривать нулевые, граничные и прочие «критические» значения исходных данных (если такие значения могут иметь место).

Тестовые данные должны включать такие комбинации исходных данных, чтобы обеспечивалось выполнение всех исполняемых операторов программы, в том числе ветвлений и циклов.

Пример (№ 169)

Требовалось написать программу, которая решает неравенство $ax + b > 0$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и неправильно написал программу.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b,x); if a = 0 then write('любое число') else if a > 0 then write('x >', -b/a) else write('x <', -b/a); end.</pre>	<pre>INPUT a, b, x IF a = 0 THEN PRINT "любое число" ELSE IF a > 0 THEN PRINT "x > ", -b/a ELSE PRINT "x < ", -b/a END</pre>	<pre>void main(void) { float a,b,x; scanf("%f%f%f",&a,&b,&x); if (a=0) printf("любое число"); else if (a>0) printf("x > %f", -b/a); else printf("x < %f", -b/a); }</pre>

Последовательно выполните три задания:

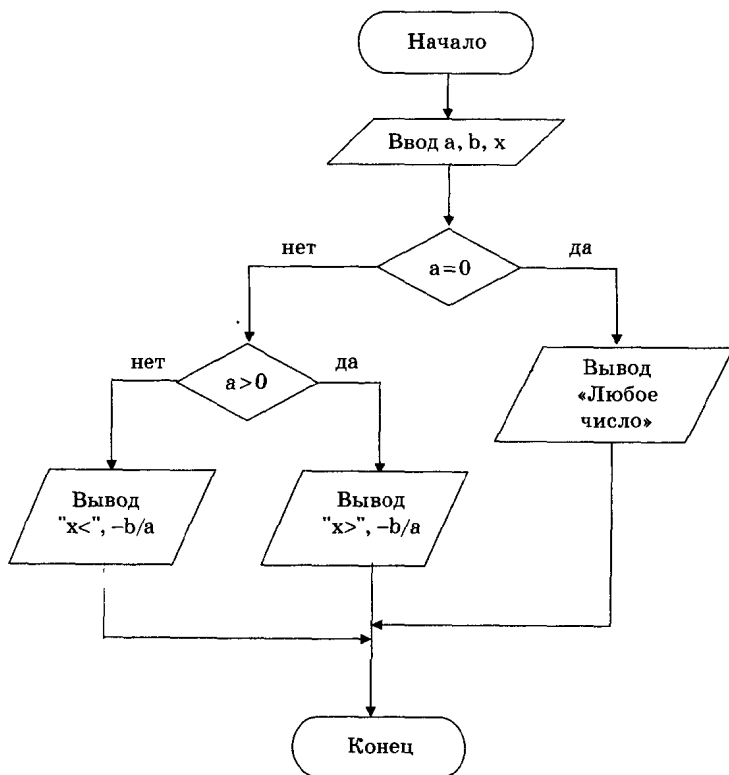
1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу.

2) Укажите, какая часть программы является лишней.

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

Решение

Рассмотрим текст программы на языке Паскаль. Составим блок-схему этой программы.



По созданной блок-схеме можно сделать следующие выводы о логических ошибках в программе.

Ввод значения x является излишним, так как впоследствии x либо заново вычисляется, либо не используется, поскольку относится к выходным, а не входным данным.

В случае $a = 0$ не анализируется знак переменной b , поэтому при любом неположительном значении b программа будет давать неверный ответ.

Один из вариантов доработки программы:

```
readln(a,b);
if a=0 then
if b>0 then
write('любое число')
else
write('нет решений')
else
if a>0 then
write('x>', -b/a)
else
write('x<', -b/a);
```

Критерии оценивания решения можно посмотреть в разделе ответов. Вкратце они сводятся к тому, что за каждый верно выполненный элемент задания (из трех указанных в условии) начисляется по одному баллу.

Практика показывает, что весьма распространенной (и весьма досадной!) является следующая ошибка учащихся: вместо требуемого в задании примера исходных данных, при которых программа работает *неправильно*, они приводят пример, когда ошибочная программа работает *правильно*, и из-за этого теряют один балл. Можно предположить, что здесь дело не только в невнимательности учеников при чтении задания, но и в том, что при изучении программирования не всегда правильно формулируется роль тестовых исходных данных в процессе разработки программы, опускается важное положение о том, что цель тестирования на этапе отладки программы не только (и не столько) продемонстрировать работоспособность программы, сколько выявить возможные содержащиеся в ней ошибки.

В задании С4 требуется написать программу для решения определенной задачи. Поскольку подробное изложение методики написания учебных программ выходит за рамки данного пособия, дадим лишь некоторые рекомендации.

Перед тем как приступить к написанию программного кода, необходимо спроектировать программу, т.е. осознать постановку задачи и разработать алгоритм решения без привязки к конкрет-

ному языку программирования. Разработанный алгоритм желательно зафиксировать в виде схемы или словесного описания.

В процессе написания программы следует активно использовать комментарии для пояснения существенных моментов алгоритма, это помогает при самопроверке и внесении исправлений в текст. Кроме того, это способствует эффективности экспертной проверки.

Текст программы (да и всех ответов на задания части С) следует стараться писать как можно разборчивее и аккуратнее, проявляя, тем самым, уважение к труду экспертов.

Развернутые ответы, примеры программ и критерии оценивания заданий С4 приведены в разделе ответов (№ 176—183).

ОТВЕТЫ

№ задания	Ответ	№ задания	Ответ	№ задания	Ответ
1	2	18	4	35	3
2	2	19	2	36	3
3	4	20	2	37	60
4	1	21	1	38	5625
5	2	22	1	39	3750
6	2	23	4	40	2400
7	3	24	1	41	3
8	1	25	3	42	4
9	1	26	1	43	2
10	1	27	4	44	4
11	2	28	3	45	2
12	3	29	6,9,18	46	4
13	1	30	7,21	47	ААСВА АВА
14	4	31	5,13,21	48	85
15	2	32	4	49	170
16	1	33	1	50	170
17	4	34	4	51	1

52.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)					
Выигрывает первый игрок, своим первым ходом он должен увеличить в 3 раза количество камней в первой куче. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучках соответственно.					
1-й ход		2-й ход		3-й ход	Пояснение
Позиция после первого хода	2-й игрок (все варианты хода)	1-й игрок (выигрышный ход)	2-й игрок (все варианты хода)	1-й игрок (один из вариантов)	
3,2	3,5	6,5	18,5	54,5	Первый игрок выигрывает на третьем ходу, после любого

1-й ход		2-й ход		3-й ход	Пояснение
По- зиция после перво- го хода	2-й игрок (все вари- анты хода)	1-й игрок (выи- грыш- ный ход)	2-й игрок (все ва- рианты хода)	1-й игрок (один из ва- риан- тов)	
3,2	3,5	6,5	9,5	27,5	ответа второго игро- ка, например, утроив число камней в самой большой кучке
			6,8	6,24	
			6,15	6,45	
	6,2	6,5	Те же варианты ходов		
	3,6	6,6	9,6	27,6	Первый игрок выи- грышает на третьем ходу, после любого ответа второго игро- ка, например, утроив число камней в самой большой кучке
			18,6	54,6	
	9,2	27,2	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.

Указания по оцениванию	Баллы
Правильное указание выигрывающего игрока и его ходов со строгим доказательством правильности (с помощью или без помощи дерева игры)	3
Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказа- тельства ее правильности	2
При наличии в представленном решении одного из пун- ктов: 1. Правильно указан выигрывающий игрок и его первый ход, рассмотрены все возможные ответы второго игрока, но неверно определены дальнейшие действия. 2. Правильно указан выигрывающий игрок и его первый ход, но описание выигрышной стратегии неполно и рас- смотрены несколько (больше одного, но не все(!) вариан- тов ответов второго игрока	1

Указания по оцениванию	Баллы
В представленном решении полностью отсутствует описание элементов выигрышной стратегии, и отсутствует анализ вариантов первого-второго ходов играющих (даже при наличии правильного указания выигрывающего игрока)	0
Максимальный балл	3

53.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)					
Выигрывает второй игрок. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры, в первой и второй кучках соответственно.					
1-й ход		2-й ход		3-й ход	Пояснение
Стартовая позиция	1-й игрок (все варианты хода)	2-й игрок (выигрышный ход)	1-й игрок (все варианты хода)	2-й игрок (один из вариантов)	
3,2	3,5	6,5	18,5	54,5	Второй игрок выигрывает на втором ходу, после любого ответа первого игрока, например, утроив число камней в самой большой кучке
			9,5	27,5	
			6,15	6,45	
			6,8	6,24	
	6,2	6,5	Те же варианты ходов		
	3,6	6,6	18,6	54,6	Второй игрок выигрывает на втором ходу, после любого ответа первого игрока, например, утроив число камней в самой большой кучке
			9,6	27,6	
	9,2	27,2	Второй игрок выигрывает ответным ходом		

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

Указания по оцениванию: см. № 52.

54.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)					
Выигрывает второй игрок. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры, в первой и второй кучках соответственно.					
	1-й ход	2-й ход	3-й ход	4-й ход	Пояснение
Стартовая позиция	1-й игрок (все варианты хода)	2-й игрок (выигршный ход)	1-й игрок (все варианты хода)	2-й игрок (один из вариантов)	
3,2	3,3	4,3	4,4	12,4	Второй игрок выигрывает на четвертом ходу, после любого ответа первого игрока, например, утроив число камней в самой большой кучке
			9,4	27,4	
			3,12	3,36	
			3,5	3,15	
	4,2	4,3	Те же варианты третьего-четвертого ходов		
	9,2	27,2	Второй игрок выигрывает ответным ходом		
	3,6	18,3	Второй игрок выигрывает ответным ходом		

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

Указания по оцениванию: см. № 52.

№ задания	Ответ	№ задания	Ответ	№ задания	Ответ
55	3	66	2212	76	3
56	2	67	121221	77	4
57	3	68	11121	78	2
58	3	69	2	79	2
59	2	70	60	80	3
60	3	71	4144	81	1
61	4	72	1,A6	82	2
62	2	73	4	83	3
63	1	74	2,A6,E7	84	2
64	1	75	2	85	4
65	121211				

86.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Введем целочисленную переменную SumProd, в которую будем заносить сумму произведений последовательных пар элементов в просмотренной части массива, и присвоим ей начальное значение 0. В цикле до конца массива: считаем произведение элементов очередной пары и прибавляем его к SumProd. По окончании цикла переменная SumProd содержит сумму произведений последовательных пар элементов массива.

Пример правильной и эффективной программы (на основе алгоритма, использующего однократный проход по массиву):

На языке Паскаль	На языке Бейсик
<pre>const N=30; var a:array[1..N] of integer; SumProd, i: integer; begin SumProd:=0; for i:=1 to N div 2 do begin SumProd:=SumProd+a[2*i-1]*a[2*i]; end; writeln(SumProd); end.</pre>	<pre>N=30 DIM i, SumProd, a(N) AS INTEGER SumProd=0 FOR i = 1 TO N/2 SumProd=SumProd+a(2*i-1)*a(2*i) NEXT i PRINT SumProd END</pre>

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение (в том числе и алгоритм, требующий двукратного прохода по массиву или создания массива произведений пар)</p> <p>Возможно использование числа 30 вместо константы. Возможно наличие отдельных синтаксических ошибок (пропущенные «;», неверная запись оператора присваивания и т.п.), не искажающих замысла автора программы</p>	2

Указания по оцениванию	Баллы
Имеется не более двух ошибок из числа следующих: 1) Не задано первое значение SumProd. 2) Не указано или неверно указано условие завершения цикла. 3) Индексная переменная в цикле не меняется. 4) Неверно расставлены операторные скобки	1
Ошибок, перечисленных выше, больше двух, или алгоритм сформулирован неверно (в частности, если хотя бы один из элементов массива умножается более одного раза)	0
Максимальный балл	2

87.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Введем целочисленную переменную MaxProd, в которую будем заносить максимальное произведение двух последовательных элементов в просмотренной части массива, и переменную MaxNum, в которую будем заносить номер первого элемента в этой паре. Первоначально в эти переменные занесем произведение первых двух элементов и номер 1 соответственно. В цикле до конца массива: проверяем, превосходит ли произведение элементов очередной пары уже найденный максимум; если да, то заносим в переменную MaxProd новое произведение, а в переменную MaxNum — номер первого элемента пары. По окончании цикла выводим значение переменной MaxNum. Пример правильной и эффективной программы (на основе алгоритма, использующего однократный проход по массиву):

На языке Паскаль	На языке Бейсик
<pre>const N = 30; var a:array[1..N] of integer; MaxProd, MaxNum, i: integer; begin MaxNum:=1; MaxProd:=a[1]*a[2]; for i:=2 to N-1 do begin if a[i]*a[i+1]>MaxProd then begin MaxNum:=i; MaxProd:=a[i]*a[i+1]; end; end; writeln(MaxNum); end.</pre>	<pre>N = 30 DIM i, MaxProd, MaxNum, a(N) AS INTEGER MaxNum=1 MaxProd=a(1)*a(2) FOR i = 2 TO N-1 IF a(i)*a(i+1)>MaxProd THEN MaxNum=i MaxProd=a(i)*a(i+1) ENDIF NEXT i PRINT MaxNum END</pre>

Указания по оцениванию	Баллы
Предложен правильный алгоритм, выдающий верное значение (в том числе и алгоритм, требующий двукратного прохода по массиву или создания массива произведений пар). Возможно использование числа 30 вместо константы. Возможно наличие отдельных синтаксических ошибок (пропущенные «;», неверная запись оператора присваивания и т.п.), не искажающих замысла автора программы	2
Имеется не более двух ошибок из числа следующих: 1) Не задано первое значение MaxNum. 2) Неверно задается первое значение MaxProd. 3) Не указано или неверно указано условие завершения цикла. 4) Индексная переменная в цикле не меняется. 5) Неверно расставлены операторные скобки	1
Ошибок, перечисленных выше, больше двух, или алгоритм сформулирован неверно	0
Максимальный балл	2

88.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
Введем целочисленную переменную MaxSum, в которую будем заносить максимальную сумму двух последовательных элементов в просмотренной части массива, и переменную MaxNum, в которую будем заносить номер первого элемента в этой паре. Первоначально в эти переменные занесем сумму первых двух элементов и номер 1 соответственно. В цикле до конца массива: проверяем, превосходит ли сумма элементов очередной пары уже найденный максимум; если да, то заносим в переменную MaxSum новую сумму, а в переменную MaxNum — номер первого элемента пары. По окончании цикла выводим значение переменной MaxNum. Пример правильной и эффективной программы (на основе алгоритма, использующего однократный проход по массиву):	
На языке Паскаль	На языке Бейсик
<pre>const N=30; var a:array[1..N] of integer; MaxSum, MaxNum, i: integer; begin MaxNum:=1; MaxSum:=a[1]+a[2]; for i:=2 to N-1 do</pre>	<pre>N=30 DIM i, MaxSum, MaxNum, a(N) AS INTEGER MaxNum=1 MaxSum=a(1)+a(2) FOR i = 2 TO N-1 IF a(i)+a(i+1)>MaxSum THEN</pre>

На языке Паскаль	На языке Бейсик	
<pre>begin if a[i]+a[i+1]>MaxSum then begin MaxNum:=i; MaxSum:=a[i]+a[i+1]; end end; writeln(MaxNum); end.</pre>	<pre>MaxNum=i MaxSum=a(i)+a(i+1) ENDIF NEXT i PRINT MaxNum END</pre>	
Указания по оцениванию		Баллы
Предложен правильный алгоритм, выдающий верное значение (в том числе и алгоритм, требующий двукратно-го прохода по массиву или создания массива сумм пар). Возможно использование числа 30 вместо константы. Возможно наличие отдельных синтаксических ошибок (пропущенные «;», неверная запись оператора присваива-ния и т.п.), не искажающих замысла автора программы		2
Имеется не более двух ошибок из числа следующих: 1) Не задано первое значение MaxNum. 2) Неверно задается первое значение MaxSum. 3) Не указано или неверно указано условие завершения цикла. 4) Индексная переменная в цикле не меняется. 5) Неверно расставлены операторные скобки		1
Ошибок, перечисленных выше, больше двух, или алго- ритм сформулирован неверно		0
Максимальный балл		2

89.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Введем и обнуляем целочисленную переменную SumNeg и целочисленную переменную NumNeg, в которые будем заносить соответственно сумму и число положительных элементов в рассмотренной части массива, и присвоим им значение 0. В цикле до конца массива: проверяем, является ли очередной элемент отрицательным. Если да, то прибавляем его к SumNeg и увеличиваем счетчик NumNeg на единицу. По окончании цикла выводим SumNeg/NumNeg, проверив предварительно, что хотя бы один отрицательный элемент в массиве есть.</p> <p>Пример правильной и эффективной программы (на основе алгоритма, использующего однократный проход по массиву):</p>

На языке Паскаль	На языке Бейсик	
<pre>Const N = 30; Var a:array [1..N] of integer; SumNeg, NumNeg, I: integer; begin SumNeg :=0; NumNeg :=0; for I := 1 to N do if a[I]<0 then begin SumNeg := SumNeg + a[I]; NumNeg := NumNeg + 1; end; if NumNeg>0 then writeln (SumNeg/NumNeg); end.</pre>	<pre>N = 30 DIM I, SumNeg, NumNeg, A(N) AS INTEGER SumNeg=0 NumNeg=0 FOR I = 1 TO N IF A(I)<0 THEN SumNeg = SumNeg + A(I) NumNeg = NumNeg + 1 ENDIF NEXT I IF NumNeg>0 THEN PRINT SumNeg/NumNeg ENDIF END</pre>	
Указания по оцениванию		Баллы
Предложен правильный алгоритм, выдающий верное значение (в том числе и алгоритм, требующий двукратного прохода по массиву). Возможно использование числа 30 вместо константы. Возможно отсутствие проверки наличия отрицательных элементов перед выдачей результатов. Возможно наличие отдельных синтаксических ошибок (пропущенные «;», неверная запись оператора присваивания и т.п.), не искажающих замысла автора программы		2
Имеется не более двух ошибок из числа следующих: 1) Не задано первое значение SumNeg или NumNeg 2) Не указано или неверно указано условие завершения цикла 3) Индексная переменная в цикле не изменяется 4) В программе на Паскале неверно расставлены операторные скобки		1
Ошибок, перечисленных выше, больше двух, или алгоритм сформулирован неверно (в частности, не выделяются отрицательные элементы), или результат работы программы не получен (не сохранен в виде переменной, либо не выведен на печать (экран), либо не возвращен при помощи команды return)		0
Максимальный балл		2

90. /1.2.4/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Пример правильного описания алгоритма на русском языке.
Заводим и обнуляем переменную MaxNeg для хранения максимального количества подряд идущих отрицательных элементов и

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

счетчик NumNeg для хранения числа отрицательных элементов в последней группе отрицательных элементов. Просматривая элементы массива, сравниваем очередной элемент с 0. Если очередной элемент массива оказывается неотрицательным, то сравниваем текущее значение счетчика NumNeg со значением переменной MaxNeg; если он больше, то заменяем значение переменной MaxNeg значением счетчика, при этом значение NumNeg обнуляется. Так повторяем до конца массива. В конце работы нужно еще раз сравнить значение счетчика со значением переменной MaxNeg и переопределить ее, если счетчик больше.

При оценке других вариантов алгоритма решения необходимо проверить, что поставленная цель достигается.

Пример правильной и эффективной программы (на основе алгоритма, использующего однократный проход по массиву):

На языке Паскаль	На языке Бейсик
<pre>const N=30; var a:array[1..N] of integer; MaxNeg, NumNeg, i: integer; begin MaxNeg:=0; NumNeg:=0; for i:=1 to N do begin if a[i]<0 then NumNeg:=NumNeg+1 else begin if NumNeg> MaxNeg then MaxNeg:=NumNeg; NumNeg:=0; end; end; end; if NumNeg> MaxNeg then MaxNeg:=NumNeg; writeln(MaxNeg); end.</pre>	<pre>N=30 DIM i, MaxNeg, NumNeg, a(N) AS INTEGER MaxNeg=0 NumNeg=0 FOR i = 1 TO N IF a(i)<0 THEN NumNeg=NumNeg+1 ELSE IF NumNeg>MaxNeg THEN MaxNeg=NumNeg END IF NumNeg=0 END IF NEXT i IF NumNeg>MaxNeg THEN MaxNeg=NumNeg END IF PRINT MaxNeg END</pre>

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение (в том числе и алгоритм, требующий двукратного прохода по массиву). Возможно использование числа 30 вместо константы. Возможно наличие отдельных синтаксических ошибок (пропущенные «;», неверная запись оператора присваивания и т.п.), не искажающих замысла автора программы</p>	2

Указания по оцениванию	Баллы
<p>Имеется не более двух ошибок из числа следующих:</p> <p>1) Не задано начальное значение MaxNeg и/или NumNeg</p> <p>2) Не указано или неверно указано условие завершения цикла</p> <p>3) Программа не выводит результат</p> <p>4) Индексная переменная в цикле не увеличивается</p> <p>5) В программе на Паскале неверно расставлены операторные скобки</p>	1
Ошибок, перечисленных выше, больше двух, или алгоритм сформулирован неверно (в частности, переменная NumNeg не обнуляется в случае перемены знака или нет проверки после завершения цикла)	0
Максимальный балл	2

91. /1.2.4/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Пример правильного описания алгоритма на русском языке. Заводим переменную MaxIncr для хранения максимального количества подряд идущих в порядке возрастания элементов и счетчик NumIncr для хранения числа элементов в последней группе возрастающих элементов. Просматривая элементы массива, сравниваем очередной элемент со следующим за ним. Если очередной элемент массива оказывается не меньше следующего, то сравниваем текущее значение счетчика со значением переменной MaxIncr; если он больше, то заменяем значение переменной MaxIncr значением счетчика, при этом значение NumIncr обнуляется. Так повторяем до конца массива. В конце работы нужно еще раз сравнить значение счетчика со значением переменной MaxIncr и переопределить ее, если счетчик больше. При оценке других вариантов алгоритма необходимо проверить, что поставленная цель достигается. Пример правильной и эффективной программы (на основе алгоритма, использующего однократный проход по массиву):</p>	
На языке Паскаль	На языке Бейсик
<pre>const N = 30; var a:array[1..N] of integer; MaxIncr, NumIncr, i: integer; begin MaxIncr:=0; NumIncr:=0; for i:=1 to N-1 do begin if a[i]<a[i+1] then NumIncr:=NumIncr+1;</pre>	<pre>N = 30 DIM i, MaxIncr, NumIncr, a(N) AS INTEGER MaxIncr=0 NumIncr=0 FOR i = 1 TO N-1 IF a(i)<a(i+1) THEN NumIncr=NumIncr+1 ELSE IF NumIncr>MaxIncr THEN</pre>

На языке Паскаль	На языке Бейсик	
<pre>else begin if NumIncr> MaxIncr then MaxIncr:=NumIncr; NumIncr:=0; end; end; if NumIncr> MaxIncr then MaxIncr:=NumIncr; writeln(MaxIncr); end.</pre>	<pre>MaxIncr=NumIncr END IF NumIncr=0 END IF NEXT i IF NumIncr>MaxIncr THEN MaxIncr=NumIncr END IF PRINT MaxIncr END</pre>	
Указания по оцениванию		Баллы
Предложен правильный алгоритм, выдающий верное значение (в том числе и алгоритм, требующий двукратного прохода по массиву). Возможно использование числа 30 вместо константы. Возможно наличие отдельных синтаксических ошибок (пропущенные «;», неверная запись оператора присваивания и т.п.), не искажающих замысла автора программы		2
Имеется не более двух ошибок из числа следующих: 1) Не задано начальное значение MaxIncr и/или NumIncr 2) Не указано или неверно указано условие завершения цикла 3) Программа не выводит результат 4) Индексная переменная в цикле не увеличивается 5) В программе на Паскале неверно расставлены операторные скобки		1
Ошибок, перечисленных выше, больше двух, или алгоритм сформулирован неверно (в частности, переменная NumIncr не обнуляется в случае обрыва роста элементов или нет проверки после завершения цикла в варианте решения, аналогичном предложенному)		0
Максимальный балл		2

№ задания	Ответ	№ задания	Ответ	№ задания	Ответ
92	4	119	3	144	1
93	2	120	3	145	2
94	3	121	2	146	1
95	1	122	2	147	4
98	2	123	3	148	1
99	3	124	2	149	1
100	4	125	3	150	1
101	1	126	2	151	4

№ задания	Ответ	№ задания	Ответ	№ задания	Ответ
102	4	127	4	152	1
103	9	128	2	153	1
104	44	129	1	154	4
105	17	130	4	155	4
106	7	131	3	156	3
107	1100	132	3	157	3
108	9	133	2	158	4
109	1100	134	4	159	1
110	10	135	2	160	4
111	1423	136	2	161	ГВЖЕДБА
112	В	137	1	162	ДАЖБЕГВ
113	А	138	3	163	ЕДАВЖБГ
114	М	139	2	164	БГЖДВЕА
115	3	140	3	165	ГВАБ
116	4	141	3	166	БАГВ
117	2	142	2	167	ББАГ
118	4	143	4	168	АГБВ

169.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1) Пример: $a = 0$ $b = -2$ $x = 1$ (значение x может быть не указано), также допустим ответ $a = 0$, $b \leq 0$, x — любое число.

2) Лишняя часть:

не нужно вводить x с клавиатуры,

верно: `readln(a,b);`

3) Возможная доработка:

`readln(a,b);`

`if a=0 then`

`if b>0 then`

`write('любое число')`

`else`

`write('нет решений')`

`else`

`if a>0 then`

`write('x>',-b/a)`

`else`

`write('x<',-b/a);`

(могут быть и другие способы доработки)

Указания по оцениванию	Баллы
Правильно выполнены все 3 пункта задания, при этом в работе (во фрагментах программ) допускается не более одной синтаксической ошибки	3
Правильно выполнены 2 пункта задания. При этом в сданной работе допускается не более двух синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)	2
Правильно выполнен только один пункт задания, при этом если это был п. 3), то в нем допускается не более трех синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)	1
Все пункты задания выполнены неверно	0
Максимальный балл	3

170.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Элементы ответа:</p> <p>1) Пример: $a = -2$ $b = 1$ $x = 1$ (значение x может быть не указано), также допустим ответ $a < 0$, b любое число, x — любое число.</p> <p>2) Лишняя часть: не надо вводить x с клавиатуры, верно: <code>readln(a,b);</code></p> <p>3) Возможная доработка: <code>readln(a,b);</code> <code>if a = 0 then</code> <code>if b > 0 then</code> <code>write('любое число')</code> <code>else</code> <code>write('нет решений')</code> <code>else</code> <code>if a > 0 then</code> <code>write('x > ', -b/a)</code> <code>else</code> <code>write('x < ', -b/a);</code> (могут быть и другие способы доработки)</p>

Указания по оцениванию см. № 169

171.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Элементы ответа:</p> <p>1) $a = 0$ $b = 0$, $x = 0$ (значение x можно не указывать), допустим ответ, что x — любое число</p>

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)

2) Лишняя часть:

не нужно вводить x с клавиатуры,верно: `readln(a,b);`3) Возможная доработка: `readln(a,b);``if a = 0 then``if b = 0 then``write('любое число')``else``write('нет решений')``else``write('x=', -b/a);`

(могут быть и другие способы доработки)

172. /2.8.1/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1) $a = 1$, $b = -1$, $x = 0$. (Значение x может быть не указано. Значения a и b могут быть любыми ненулевыми числами с разными знаками. Также допустим ответ, что программа работает неправильно при любых ненулевых a и b , имеющих разные знаки.)

2) Лишняя часть:

не нужно вводить x с клавиатуры,верно: `readln(a,b);`

3) Возможная доработка:

`readln(a,b);``if a = 0 then``if b = 0 then write('любое число')``else write('нет решений')``else``if b/a > 0 then``write('x=', b/a, ' или x=', -b/a)``else``if b=0 then write('x=0')``else write('нет решений');`

(могут быть и другие способы доработки).

При оценке других вариантов доработки программы нужно проверять, что поставленная цель достигается

Указания по оцениванию	Баллы
Правильно выполнены все 3 пункта задания, при этом в работе (во фрагментах программ) допускается не более одной синтаксической ошибки	3

Указания по оцениванию	Баллы
Правильно выполнены 2 пункта задания. При этом в сданной работе допускается не более двух синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)	2
Правильно выполнен только один пункт задания, при этом, если это был п.3), то в нем допускается не более трех синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)	1
Все пункты задания выполнены неверно	0
Максимальный балл	3

173. /2.8.1/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1) $a = -1$ $b = 1$ $x = 0$. (Значение x может быть не указано. Значение a может быть любым отрицательным числом, значение b — любым положительным. Также допустим ответ, что программа работает неправильно при любых положительных b и отрицательных a .)

2) Лишняя часть:

не нужно вводить x с клавиатуры,

верно: `readln(a,b);`

3) Возможная доработка:

`readln(a,b);`

`if b>0 then`

`if a>0 then`

`write('x>',a,' или x<0')`

`else`

`write('x<',a,' или x>0')`

`else`

`if a>0 then`

`write('0<x<',a)`

`else`

`write(a,'<x<0');`

(могут быть и другие способы доработки).

При оценке других вариантов доработки программы нужно проверять, что поставленная цель достигается

Указания по оцениванию – см. № 171

174. /2.8.1/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1) $a = 1, b = -1, x = 0$. (Значение x может быть не указано. Значение b может быть любым отрицательным числом, значение a — любым положительным. Также допустим ответ, что программа работает неправильно при любых положительных a и отрицательных b .)

2) Лишняя часть:

не нужно вводить x с клавиатуры,

верно: `readln(a,b);`

3) Возможная доработка:

`readln(a,b);`

`if a=0 then`

`if b>0 then`

`write('нет решений')`

`else`

`write('x>0 или x<0')`

`else`

`if b>0 then`

`write(-a,'<x<0')`

`else`

`write('x>0 или x<',-a);`

(могут быть и другие способы доработки).

При оценке других вариантов доработки программы нужно проверять, что поставленная цель достигается

Указания по оцениванию — см. № 171

175. /2.8.1/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1) $a = 1, b = -1, x = 0$. (Значение x может быть не указано. Значения a и b могут быть любыми ненулевыми числами с разными знаками. Также допустим ответ, что программа работает неправильно при любых ненулевых a и b , имеющих разные знаки.)

2) Лишняя часть:

не нужно вводить x с клавиатуры,

верно: `readln(a,b);`

3) Возможная доработка:

`readln(a,b);`

`if a = 0 then`

`if b = 0 then write('любое число')`

`else write('нет решений')`

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

```
else
if  $b/a > 0$  then
write('x=', $b/a$ , ' или x=', $-b/a$ )
else
if  $b=0$  then write('x=0')
else write('нет решений');
```

(могут быть и другие способы доработки).
При оценке других вариантов доработки программы нужно проверять, что поставленная цель достигается

Указания по оцениванию см. № 169

176.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Программа верно читает входные данные, запоминая фамилии, имена и сумму баллов в массиве записей (или в нескольких массивах), сразу или за дополнительный просмотр подсчитывая три лучшие по величине суммы баллов (так как количество экзаменов у всех учащихся одинаковое, лучший средний балл соответствует лучшей сумме баллов). Затем за дополнительный просмотр этого массива распечатывается информация о тех учениках, которые набрали в сумме баллов не меньше третьей по величине суммы. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, все ученики набрали различный средний балл)

Пример правильной и эффективной программы на языке Паскаль:

```
var p:array[1..100] of record
    name:string,
    sum:integer;
end,
c:char;
i,j,N,s1,s2,s3,m:integer;
begin
readln(N);
for i:=1 to N do
begin
p[i].name:= "";
repeat
read(c);
```

```

p[i].name:=p[i].name+c
until c=' '; {считана фамилия}
repeat
  read(c);
  p[i].name:=p[i].name+c
until c=' '; {считано имя}
p[i].sum:=0;
for j:=1 to 3 do
begin
  read(m);
  p[i].sum:=p[i].sum+m
end; {подсчитана сумма баллов}
readln;
end;
s1:=0; s2:=0; s3:=0;
for i:=1 to N do
begin
  if p[i].sum>s1 then
  begin
    s3:=s2; s2:=s1;
    s1:=p[i].sum
  end else
  if p[i].sum>s2 then
  begin
    s3:=s2; s2:=p[i].sum
  end else
  if p[i].sum>s3 then s3:=p[i].sum;
end;
for i:=1 to N do
  if p[i].sum>=s3 then writeln(p[i].name);
end.

```

Пример правильной программы на языке Бейсик:

```

DIM i, j, n, s1, s2, s3, sum(100) AS INTEGER
DIM s AS STRING
DIM nm(100) AS STRING
INPUT n
FOR j = 1 TO n
  LINE INPUT s
  c$ = MID$(s, 1, 1)
  i = 1
  WHILE NOT (c$ = « »)
    i = i + 1
    c$ = MID$(s, i, 1)
  
```

```

WEND
i=i+ 1
c$=MID$(s, i, 1)
WHILE NOT (c$ =« «)
i=i+ 1
c$=MID$(s, i, 1)
WEND
nm(j)=MID$(s, 1, i)
sum(j)=ASC(MID$(s, i+ 1, 1)) - ASC(«0»)
sum(j)=sum(j)+(ASC(MID$(s,i+3,1))-ASC(«0»))
sum(j)=sum(j)+(ASC(MID$(s,i+5,1))-ASC(«0»))
NEXT j
s1=0 s2=0 s3=0
FOR j=1 TO n
IF sum(j) > s1 THEN
s3=s2: s2=s1
s1=sum(j)
ELSE
IF sum(j) > s2 THEN
s3=s2: s2=sum(j)
ELSE
IF sum(j) > s3 THEN s3=sum(j)
END IF
END IF
NEXT j
FOR j=1 TO n
IF sum(j) >=s3 THEN PRINT nm(j)
NEXT j
END

```

Указания по оцениванию	Баллы
Программа работает верно, т.е. корректно выделяет из входных данных оценки, ищет три лучшие суммы баллов и распечатывает список учеников, набравших эти суммы. Допускается наличие в тексте программы одной синтаксической ошибки	4
Программа в целом работает верно, но содержит по крайней мере две из следующих неточностей (нерациональностей): сохраняются не суммы баллов (средние баллы), а сами баллы, и суммы несколько раз заново перевычисляются; явно вычисляются средние баллы, что приводит к сравнению вещественных чисел; при нахождении трех максимальных значений элементы массива переставляются местами; при печати с каждым из трех максимальных элементов производятся сравнения. Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено	3

Указания по оцениванию	Баллы
зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	
Программа в целом работает верно, но выводит только трех лучших учеников, даже если кто-то еще сдал экзамены не хуже. Возможно, в реализации алгоритма содержатся 1—2 ошибки (используется знак “<” вместо “>”, “or” вместо “and” и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие до пяти синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	2
Программа неверно работает при некоторых входных данных и, возможно, содержит ошибку в алгоритме поиска трех максимальных элементов. Допускается до 4 разных ошибок в ходе решения задачи, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	1
Задание выполнено неверно	0
<i>Максимальный балл</i>	4

177.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
Программа верно читает входные данные, сразу запоминая в массиве только фамилии и времена окончания хранения багажа тех пассажиров, которые должны освободить ячейки в ближайшие 2 ч. Время при считывании удобно перевести в минуты и в этом же виде хранить и сравнивать. Затем полученный массив времен сортируется по неубыванию любым алгоритмом сортировки, параллельно переставляются и элементы массива с фамилиями (возможно использование одного массива записей, состоящих из двух полей). Печатаются элементы массива фамилий в полученном в результате сортировки порядке	

Пример правильной и эффективной программы на языке Паскаль:

```

type pp=record
    name:string[20];
    time:integer;
end;
var
    p.array[1..1000] of pp;
    q:pp,
    c,c1:char;
    i,j,N,time1:integer;
begin
    read(c,c1); {читаны часы текущего времени}
    time1:=60*((ord(c)-ord('0'))*10+ ord(c1)-ord('0')),
    readln(c,c,c1); {пропущено двоеточие, и считаны минуты}
    time1:=time1+(ord(c)-ord('0'))*10+ord(c1)-ord('0');
    readln(N);
    j:=1;
    for i:=1 to N do
    begin
        p[j].name:="";
        repeat
            read(c);
            p[j].name:=p[j].name+c
        until c=' '; {считана фамилия}
        read(c,c1); {читаны часы первого времени}
        p[j].time:=60*((ord(c)-ord('0'))*10+ ord(c1)-ord('0'));
        readln(c,c,c1); {пропущено двоеточие, и считаны минуты}
        p[j].time:=p[j].time+(ord(c)-ord('0'))*10+ord(c1)-ord('0');
        if (p[j].time>=time1)and(p[j].time<=time1+120)then
            j:=j+1; {данные занесены в массив}
        end;
        N:=j-1;
    for i:=1 to N-1 do {сортируем данные}
        for j:=1 to N-i do
            if p[j].time>p[j+1].time then
                begin
                    q:=p[j];
                    p[j]:=p[j+1],
                    p[j+1]:=q,
                end;
    for i:=1 to N do
        writeln(p[i].name)
    end

```

Пример правильной программы на языке Бейсик:

```

DIM t(1000) AS INTEGER
DIM m(1000) AS STRING * 20
DIM s AS STRING
DIM nm AS STRING
LINE INPUT s
time1 =(ASC(MID$(s, 1, 1)) - ASC(«0»)) * 60 * 10
time1 =time1+ (ASC(MID$(s, 2, 1)) - ASC(«0»)) * 60
time1 =time1+ (ASC(MID$(s, 4, 1)) - ASC(«0»)) * 10
time1 =time1+ (ASC(MID$(s, 5, 1)) - ASC(«0»))
INPUT N
k =0
FOR j =1 TO N
LINE INPUT s
c$ =MID$(s, 1, 1)
i =1
WHILE NOT (c$ =« »)
i =i+ 1
c$ =MID$(s, i, 1)
WEND
nm =MID$(s, 1, i)
time2 =(ASC(MID$(s, i+ 1, 1)) - ASC(«0»)) * 60 * 10
time2 =time2+ (ASC(MID$(s, i+ 2, 1)) - ASC(«0»)) * 60
time2 =time2+ (ASC(MID$(s, i+ 4, 1)) - ASC(«0»)) * 10
time2 =time2+ (ASC(MID$(s, i+ 5, 1)) - ASC(«0»))
IF time2 >=time1 AND time2 <=time1 + 120 THEN
k =k+ 1
t(k) =time2
m(k) =nm
END IF
NEXT j
FOR i =1 TO k - 1
FOR j =1 TO k - i
IF t(j) > t(j+ 1) THEN
time2 =t(j): nm =m(j)
t(j) =t(j+ 1) m(j) =m(j+ 1)
t(j+ ) =time2: m(j+ 1) =nm
END IF
NEXT j
NEXT i
FOR i =1 TO k
PRINT m(i)
NEXT i
END

```

Указания по оцениванию	Баллы
<p>Программа работает верно и эффективно, т.е. корректно выделяет из входных данных время, запоминает фамилии пассажиров и время выдачи багажа только тех, которые должны забрать его в ближайшие 2 ч. Фамилии этих пассажиров верно сортируются согласно временам выдачи багажа, а затем печатаются. Допускается наличие в тексте программы одной синтаксической ошибки</p>	4
<p>Программа в целом работает верно, но содержит по крайней мере две из следующих нерациональностей: сохраняются фамилии и времена для всех пассажиров, время не переводится в минуты, сортируются все фамилии, а при печати анализируется, какие из них допустимые. Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных</p>	3
<p>Программа в целом работает верно, но не всегда верно определяет допустимость времени или некорректно работает в случае отсутствия допустимых времен. Возможно, в реализации алгоритма содержатся 1–2 ошибки (используется знак “<” вместо “>”, “or” вместо “and” и т.д.). Возможно, некорректно организовано считывание входных данных. Допускается наличие до пяти синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных</p>	2
<p>Программа неверно работает при некоторых входных данных и, возможно, содержит ошибку в сортировке, или времена верно сортируются, а соответствующие им фамилии — нет. Допускается до 4 различных ошибок в ходе решения задачи, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово</p>	1

Указания по оцениванию	Баллы
языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	1
Задание выполнено неверно	0
<i>Максимальный балл</i>	4

178.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Программа читает все входные символы до точки один раз, подсчитывая в массиве, хранящем 26 целых чисел, количество каждой из букв. Сами входные символы при этом не запоминаются. В дополнительный массив, состоящий из 26 символов, заносятся буквы от “a” до “z”. Затем элементы первого массива сортируются любым алгоритмом сортировки по невозрастанию, параллельно переставляются и элементы второго массива (возможно использование одного массива записей, состоящих из двух полей). При этом элементы с равным числом вхождений символов местами не меняются. Печатаются элементы второго из отсортированных массивов, количество которых больше 0

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая (например, для строк, состоящих не более чем из 255 символов)

Пример правильной и эффективной программы на языке Паскаль:

```
var a:array[0..25] of integer;
    m:array[0..25] of 'a'..'z';
    c: char;
    i, j, k: integer;
begin
  for i:=0 to 25 do
    begin
      a[i]:=0;
      m[i]:=chr(ord('a')+i)
    end;
  read(c);
  while c < >' ' do
    begin
      a[ord(c)-ord('a')] :=a[ord(c)-ord('a')] + 1;
      read(c);
    end;
```

```

for i:= 1 to 25 do
for j :=0 to 24 do
if a[j] < a[j+1] then
begin
k:=a[j]; c:=m[j];
a[j]:=a[j+1]; m[j]:=m[j+1];
a[j+1]:=k; m[j+1]:=c
end;
i:=0,
while (i < 26) and (a[i] > 0) do
begin
write(m[i]);
i = i + 1
end;
writeln
end.

```

Пример правильной программы на языке Бейсик:

```

DIM i, j, k, a(26) AS INTEGER
DIM s(26) AS STRING * 1
FOR i = 1 TO 26
a(i) = 0
s(i) = CHR$(ASC(«a») + i - 1)
NEXT
INPUT c$
DO WHILE NOT (c$ = «.»)
a(ASC(c$) - ASC(«a») + 1) = a(ASC(c$) - ASC(«a») + 1) + 1
INPUT c$
LOOP
FOR j = 1 TO 25
FOR i = 1 TO 25
IF a(i) < a(i+ 1) THEN
k = a(i)
c$ = s(i)
a(i) = a(i+ 1)
s(i) = s(i+ 1)
a(i+ 1) = k
s(i+ 1) = c$
END IF
NEXT
NEXT
i = 1
DO WHILE i < 27 AND a(i) > 0
PRINT s(i);
i = i + 1
LOOP
END

```

Указания по оцениванию	Баллы
Программа работает верно для любых входных данных произвольного размера и строит решение, не сохраняя входные данные в строке или массиве символов. Программа просматривает входные данные один раз, в тексте программы каждая английская буква в отдельности не анализируется. Допускается наличие в тексте программы одной синтаксической ошибки	4
Программа работает верно, но входные данные запоминаются в массиве символов или строке, или входные данные считываются несколько раз. Возможно, каждая буква обрабатывается явным образом (26 операторов IF с использованием многоточия при записи программы или оператор CASE, содержащий 26 вариантов). Возможно, после сохранения входных данных для каждой буквы от “a” до “z” они заново просматриваются и подсчитывается количество вхождений соответствующей буквы. В программе присутствуют вложенные циклы (один по входным данным, второй — по буквам алфавита или их номерам, он может быть заменен оператором CASE или 26 операторами IF). Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	3
Программа в целом работает верно, эффективно или нет, но в реализации алгоритма содержатся 1–2 ошибки (используется знак “>” вместо “<”, выход за границу массива и т.п.), в том числе в алгоритме сортировки или при распечатке результата. Возможно, буквы, встречающиеся с одинаковой частотой, выводятся не по алфавиту. Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	2
Программа, возможно, неверно работает при некоторых входных данных. Возможно, программа выводит в том числе и буквы, которые не встречаются во вход-	1

Указания по оцениванию	Баллы
ных данных или содержит другие ошибки в выводе ответа. Допускается до 4 разных ошибок в реализации алгоритма, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	1
Задание выполнено неверно	0
<i>Максимальный балл</i>	4

179. /2.8.3/

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
Программа верно читает входные данные, не запоминая их все, а сразу подсчитывая в массиве, хранящем 12 целых чисел, количество учащихся в каждой из параллелей. Затем с использованием этого массива ищется параллель с максимальным числом учеников. За дополнительный просмотр этого массива распечатывается информация об искомым параллелях. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, параллель с максимальным количеством учеников единственна).

Пример правильной и эффективной программы на языке Паскаль:

```

var pc.array[1..12] of integer;
    p.1. 12,
    class:string[3];
    c:char,
    max, i, N:integer,
begin
  readln(N);
  for i:=1 to 12 do
    pc[i]:=0;
  for i =1 to N do
    begin
      repeat
        read(c)
      until c=' '; {считана фамилия}
      repeat
        read(c)

```

```

until c=' ', {считано имя}
readln(class),
{определяем номер параллели}
if length(class)=2 then
  p:=ord(class[1])-ord('0') else
  p:=(ord(class[1])-ord('0'))*10+
    ord(class[2])-ord('0');
  pc[p]:=pc[p]+1;{учитываем ученика этой параллели}
end;
max:=0;
for i=1 to 12 do
  if pc[i]>max then max:=pc[i];
writeln('Максимум учеников в параллели:',max);
for i:=1 to 12 do
  if pc[i]=max then
    write(i, ' ');
  readln
end.

```

Пример правильной программы на языке Бейсик:

```

DIM i, j, p, n, max, pc(12) AS INTEGER
DIM m(12)
DIM s AS STRING
FOR i = 1 TO 12
  pc(i) = 0
NEXT i
INPUT n
FOR j = 1 TO n
  LINE INPUT s
  c$ = MID$(s, 1, 1)
  i = 1
  WHILE NOT (c$ = " ")
    i = i + 1
    c$ = MID$(s, i, 1)
  WEND
  i = i + 1
  c$ = MID$(s, i, 1)
  WHILE NOT (c$ = " ")
    i = i + 1
    c$ = MID$(s, i, 1)
  WEND
  s = MID$(s, i + 1, 3)
  IF MID$(s, 2, 1) >= "0" AND MID$(s, 2, 1) <= "2" THEN
    p = (ASC(MID$(s, 1, 1)) - ASC("0")) * 10 +
      ASC(MID$(s, 2, 1)) - ASC("0")
  ELSE

```

```

p = ASC(MID$(s, 1, 1)) - ASC("0")
END IF
pc(p) = pc(p) + 1
NEXT j
max = 0
FOR i = 1 TO 12
  IF pc(i) > max THEN max = pc(i)
NEXT i
PRINT "Max = "; max
FOR i = 1 TO 12
  IF pc(i) = max THEN PRINT i, " ";
NEXT i
END

```

Указания по оцениванию	Баллы
Программа работает верно, т.е. корректно выделяет из входных данных номер параллели, не содержит вложенных циклов (от 1 до N и от 1 до 12 или от 1 до 26), в тексте программы не анализируется каждая параллель в отдельности. Допускается наличие в тексте программы одной пунктуационной ошибки	4
Программа работает верно, но содержит вложенные циклы (от 1 до 12 и от 1 до N) или обрабатывает каждую параллель явным образом (12 операторов IF или оператор CASE, содержащий 12 вариантов). Возможно, сохраняет все входные данные в массиве учеников. Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	3
Программа работает в целом верно, но выводит значение только одной параллели с максимальным числом учащихся. Возможно, в реализации алгоритма содержатся 1–2 ошибки (используется знак "<" вместо ">", "or" вместо "and" и т.п.). Допускается наличие до пяти синтаксических ошибок	2
Программа неверно работает при некоторых входных данных и, возможно, содержит ошибку в алгоритме поиска максимума или в выделении номера параллели из строки входных данных. Допускается наличие до семи синтаксических ошибок	1

Указания по оцениванию	Баллы
Задание выполнено неверно	0
Максимальный балл	4

180. /2.8.3/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Программа читает входные данные, сразу подсчитывая минимальную длину встречающихся слов. За второй проход исходных данных производится замена букв латинского алфавита и печать расшифрованного сообщения. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, для строчных английских букв и без циклического сдвига)

Пример правильной и эффективной программы на языке Паскаль:

```

var f:boolean;
    i, k, min :integer,
    c,cnew:char;
    s:string;
begin
  s:="";
  min:=250; k:=0;
  f:=false;
  repeat
    read(c);
    s:=s+c,
    if f then {слово началось}
      if c in ['a'..'z','A'..'Z']
        then k:=k+1
        else begin
              if k<min then min:=k,
              f:=false
            end
      else {f=false}
        if c in ['a'..'z','A'..'Z']
          then begin f:=true; k:=1 end
    until c='.';
    for i:=1 to length(s) do
      begin
        cnew:=chr(ord(s[i])+min);
        case s[i] of

```

```

'a'. 'z':if cnew>'z' then write(chr(ord(cnew)-26))
      else write(cnew);
'A'..'Z':if cnew>'Z' then write(chr(ord(cnew)-26))
      else write(cnew);
      else write(s[i])
end;
end,
readln
end.

```

Пример правильной программы на языке Бейсик:

```

DIM i, j, min, k, f, a(26) AS INTEGER
DIM s AS STRING
INPUT s
i = 1
k = 0
min = 250
f = 0
WHILE NOT (MID$(s, i, 1) = ".")
  c$ = MID$(s, i, 1)
  IF f = 1 THEN
    IF (c$ >= "A") AND (c$ <= "Z") OR
       (c$ >= "a") AND (c$ <= "z") THEN
      k = k + 1
    ELSE IF k < min THEN min = k
      f = 0
    END IF
  ELSE
    IF (c$ >= "A") AND (c$ <= "Z") OR
       (c$ >= "a") AND (c$ <= "z") THEN
      f = 1: k = 1
    END IF
  END IF
  i = i + 1
WEND
IF k < min THEN min = k
FOR j = 1 TO i
  cnew$ = CHR$(ASC(MID$(s, j, 1)) + min)
  IF (MID$(s, j, 1) >= "a") AND (MID$(s, j, 1) <= "z") THEN
    IF cnew$ > "z" THEN
      PRINT (CHR$(ASC(cnew$) - 26));
    ELSE PRINT cnew$;
    END IF
  ELSE
    IF (MID$(s, j, 1) >= "A") AND (MID$(s, j, 1) <= "Z") THEN

```



```

IF cnew$ > "Z" THEN
PRINT (CHR$(ASC(cnew$) - 26)),
ELSE PRINT cnew$,
END IF
ELSE PRINT MID$(s, j, 1);
END IF
END IF
NEXT j
END

```

Указания по оцениванию	Баллы
Программа работает верно, т.е. правильно расшифровывает сообщение, не меняя регистра английских букв, не содержит вложенных циклов (один – по количеству букв во входных данных, второй – по латинским буквам), в тексте программы не анализируется каждая буква в отдельности (if c = 'a' then, if c = 'b' then ...). Допускается наличие в тексте программы одной пунктуационной ошибки	4
Программа работает верно, но может содержать вложенные циклы или обрабатывает каждую букву явным образом (26 или 52 оператора IF или оператор CASE, содержащий 26 или 52 вариантов). Допускается наличие от одной до трех различных синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	3
Программа работает в целом верно, но неправильно вычисляет букву при необходимости зацикливания алфавита или сдвиг производится не в том направлении. Возможно, в реализации алгоритма содержатся 1–2 ошибки (используется знак "<" вместо ">", "or" вместо "and" и т.п.). Допускается наличие от одной до пяти различных синтаксических ошибок	2
Программа неверно работает при некоторых входных данных (например, со строчными буквами или текстами, содержащими символы, отличные от латинских букв), возможно, содержит ошибку в алгоритме поиска минимальной длины слова, в результате которой не всегда правильно находится величина сдвига. Допускается наличие от одной до семи различных синтаксических ошибок	1
Задание выполнено неверно	0
<i>Максимальный балл</i>	4

181. /2.8.3/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Программа считывает входные данные, сразу подсчитывая в массиве, хранящем 12 вещественных чисел, сумму температур в каждом из месяцев. Затем с использованием этого массива ищется максимальная среднемесячная температура. За дополнительный просмотр среднемесячных температур (их можно как запомнить в массиве, так и вычислить заново) распечатывается информация об искомым месяцах. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, месяц с максимальной температурой единственен)

Пример правильной и эффективной программы на языке Паскаль:

```
const d:array[1..12] of integer =
  (31,28,31,30,31,30,31,31,30,31,30,31);
var m:array[1..12] of real;
    max,t:real;
    i,j:integer;
    c1,c2:char;
begin
  for j:=1 to 12 do
    m[j]:=0;
  for i:=1 to 365 do
    begin
      readln(c1,c1,c1,c1,c2,t);
      j:=(ord(c1)-ord('0'))*10+
        ord(c2)-ord('0');
      m[j]:=m[j]+t
    end;
  max:=m[1]/d[1];
  for j:=2 to 12 do
    if m[j]/d[j] > max then
      max:=m[j]/d[j];
  for j:=1 to 12 do
    if abs(m[j]/d[j]-max) < 0.0001
    then writeln(j,' ',m[j]/d[j]:0:1)
  end.
```

Пример правильной программы на языке Бейсик:

```
DATA 31,29,31,30,31,30,31,31,30,31,30,31
DIM i, j, d(12) AS INTEGER
DIM m(12)
DIM dat AS STRING * 5
```

```

FOR i = 1 TO 12
  m(i) = 0
  READ d(i)
NEXT i
FOR i = 1 TO 365
  INPUT dat, t
  j = (ASC(MID$(dat, 4, 1)) - ASC("0")) * 10 +
    ASC(MID$(dat, 5, 1)) - ASC("0")
  m(j) = m(j) + t
NEXT i
max = m(1) / d(1)
FOR j = 2 TO 12
  IF m(j) / d(j) > max THEN max = m(j) / d(j)
NEXT j
FOR j = 1 TO 12
  IF ABS(m(j) / d(j) - max) < .0001 THEN
    PRINT j; « »;
    PRINT USING «##.##»; m(j) / d(j)
  END IF
NEXT j
END
    
```

Указания по оцениванию	Баллы
<p>Программа работает верно, т.е. определяет все месяцы, в которых среднемесячная температура максимальна, не содержит вложенных циклов (от 1 до 365 и от 1 до 12), в тексте программы не анализируется каждый месяц в отдельности (if m=1 then, if m=2 then...). При вычислении среднемесячных температур допустимо использование оператора CASE или 3-х операторов IF, для учета количества дней в том или ином месяце. Допускается наличие в тексте программы одной пунктуационной ошибки</p>	4
<p>Программа работает верно, но содержит вложенные циклы (от 1 до 12 и от 1 до 365) или обрабатывает каждый месяц явным образом при считывании данных (12 операторов IF или оператор CASE, содержащий 12 вариантов). Возможно, сохраняет все входные данные в массиве. Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных</p>	3

Указания по оцениванию	Баллы
Программа работает в целом верно, но выводит значение только одного месяца с максимальной температурой. Возможно, в реализации алгоритма содержатся 1–2 ошибки (используется знак “<” вместо “>”, “or” вместо “and” и т.п.). Допускается наличие до пяти различных синтаксических ошибок	2
Программа неверно работает при некоторых входных данных и, возможно, содержит ошибку в алгоритме поиска максимума или среднемесячной температуры или в выделении номера месяца и температуры дня из строки входных данных. Допускается наличие до семи синтаксических ошибок	1
Задание выполнено неверно	0
<i>Максимальный балл</i>	4

182. /2.8.3/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Программа читает все входные символы до точки один раз, подсчитывая в массиве, хранящем 26 целых чисел, количество каждой из букв. Сами входные символы при этом не запоминаются. В дополнительный массив, состоящий из 26 символов, заносятся буквы от “a” до “z”. Затем элементы первого массива сортируются по неубыванию любым алгоритмом сортировки, параллельно переставляются и элементы второго массива (возможно использование одного массива записей, состоящих из двух полей). При этом элементы с равным числом вхождений символов местами не меняются. Во втором из отсортированных массивов пропускаются элементы, количество которых равно 0, остальные элементы печатаются подряд.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая (например, для строк, состоящих не более чем из 255 символов).

Пример правильной и эффективной программы на языке Паскаль:

```
var a:array[0..25] of integer;
    m:array[0..25] of 'a'..'z';
    c: char;
    i, j, k: integer;
begin
  for i:=0 to 25 do
    begin
```

```

a[i]:=0;
m[i].:=chr(ord('a')+i)
end;
read(c);
while c<>'.' do
begin
a[ord(c)-ord('a')] := a[ord(c)-ord('a')] + 1,
read(c);
end,
for i:=1 to 25 do
for j := 0 to 24 do
if a[j] > a[j+1] then
begin
k:=a[j]; c:=m[j],
a[j]:=a[j+1]; m[j]:=m[j+1];
a[j+1]:=k; m[j+1]:=c
end;
i:=0;
while a[i]=0 do i:=i+1;
for j:=i to 25 do
write(m[j]);
writeln
end

```

Пример правильной и эффективной программы на языке Бейсик:

```

DIM i, j, k, a(26) AS INTEGER
DIM s(26) AS STRING * 1
FOR i = 1 TO 26
a(i) = 0
s(i) = CHR$(ASC("a") + i - 1)
NEXT
INPUT c$
DO WHILE NOT (c$ = «.»)
a(ASC(c$) - ASC("a") + 1) = a(ASC(c$) - ASC("a") + 1) + 1
INPUT c$
LOOP
FOR j = 1 TO 25
FOR i = 1 TO 25
IF a(i) > a(i + 1) THEN
k = a(i)
c$ = s(i)
a(i) = a(i + 1)
s(i) = s(i + 1)

```

```

a(i + 1) = k
s(i + 1) = c$
END IF
NEXT
NEXT
i = 1
DO WHILE a(i) = 0
  i = i + 1
LOOP
FOR j = i TO 26
PRINT s(j);
NEXT
END

```

Указания по оцениванию	Баллы
<p>Программа работает верно для любых входных данных произвольного размера и строит решение, не сохраняя входные данные в строке или массиве символов. Программа просматривает входные данные один раз, в тексте программы не анализируется каждая английская буква в отдельности. Допускается наличие в тексте программы одной синтаксической ошибки</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве символов или строке, или входные данные считываются несколько раз. Возможно, каждая буква обрабатывается явным образом (26 операторов IF с использованием многоточия при записи программы или оператор CASE, содержащий 26 вариантов). Возможно, после сохранения входных данных для каждой буквы от “a” до “z” они просматриваются заново и подсчитывается количество вхождений соответствующей буквы. В программе присутствуют вложенные циклы (один по входным данным, второй — по буквам алфавита или их номерам, он может быть заменен оператором CASE или 26 операторами IF). Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных</p>	3

Указания по оцениванию	Баллы
Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержатся 1–2 ошибки (используется знак “<” вместо “>”, выход за границу массива и т.п.), в том числе в алгоритме сортировки. Возможно, буквы, встречающиеся с одинаковой частотой, выводятся не по алфавиту. Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных	2
Программа, возможно, неверно работает при некоторых входных данных. Возможно, программа выводит в том числе и буквы, которые во входных данных не встречаются, или содержит другие ошибки в выводе ответа. Допускается до 4 различных ошибок в реализации алгоритма, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных.	1
Задание не выполнено или выполнено неверно	0
<i>Максимальный балл</i>	4

183. /2.8.3/

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
Программа верно читает входные данные, запоминая фамилии, имена и сумму баллов в массиве записей (или в нескольких массивах), сразу или за дополнительный просмотр подсчитывая три худших по величине суммы баллов (так как количество экзаменов у всех учеников одинаковое, лучший средний балл соответствует лучшей сумме баллов). Затем за дополнительный просмотр этого массива распечатывается информация о тех учениках, которые набрали в сумме баллов не больше третьей по величине суммы. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, все ученики набрали различный средний балл).

Пример правильной и эффективной программы на языке Паскаль:

```

var p:array[1..100] of record
    name:string;
    sum integer;
end;
c:char;
i,j,N,s1,s2,s3,m:integer;
begin
  readln(N);
  for i:=1 to N do
  begin
    p[i].name:='';
    repeat
      read(c);
      p[i].name:=p[i].name+c
    until c=' '; {считана фамилия}
    repeat
      read(c);
      p[i].name:=p[i].name+c
    until c=' '; {считано имя}
    p[i].sum:=0;
    for j:=1 to 3 do
    begin
      read(m);
      p[i].sum:=p[i].sum+m
    end; {подсчитана сумма баллов}
    readln,
  end,
  s1:=20; s2:=20; s3:=20;
  for i:=1 to N do
  begin
    if p[i].sum<s1 then
    begin
      s3:=s2; s2:=s1;
      s1:=p[i].sum
    end else
    if p[i].sum<s2 then
    begin
      s3:=s2; s2:=p[i].sum
    end else
    if p[i].sum<s3 then s3:=p[i].sum;
  end,
  for i:=1 to N do
    if p[i].sum<=s3 then writeln(p[i].name);
  end.

```


Пример правильной программы на языке Бейсик:

```

DIM i, j, n, s1, s2, s3, sum(100) AS INTEGER
DIM s AS STRING
DIM nm(100) AS STRING
INPUT n
FOR j = 1 TO n
LINE INPUT s
c$ = MID$(s, 1, 1)
i = 1
WHILE NOT (c$ = " ")
i = i + 1
c$ = MID$(s, i, 1)
WEND
i = i + 1
c$ = MID$(s, i, 1)
WHILE NOT (c$ = " ")
i = i + 1
c$ = MID$(s, i, 1)
WEND
nm(j) = MID$(s, 1, i)
sum(j) = ASC(MID$(s, i + 1, 1)) - ASC("0")
sum(j)=sum(j)+(ASC(MID$(s,i+3,1))-ASC("0"))
sum(j)=sum(j)+(ASC(MID$(s,i+5,1))-ASC("0"))
NEXT j
s1 = 20 s2 = 20: s3 = 20
FOR j = 1 TO n
IF sum(j) < s1 THEN
s3 = s2: s2 = s1
s1 = sum(j)
ELSE
IF sum(j) < s2 THEN
s3 = s2: s2 = sum(j)
ELSE
IF sum(j) < s3 THEN s3 = sum(j)
END IF
END IF
NEXT j
FOR j = 1 TO n
IF sum(j) <= s3 THEN PRINT nm(j)
NEXT j
END.

```

Указания по оцениванию	Баллы
Программа работает верно, т.е. корректно выделяет из входных данных оценки, ищет три худшие суммы баллов и распечатывает учеников, набравших эти суммы. Допускается наличие в тексте программы одной синтаксической ошибки	4

Указания по оцениванию	Баллы
<p>Программа работает в целом верно, но содержит по крайней мере две из следующих неточностей (нерацональностей): сохраняются не суммы баллов (средние баллы), а сами баллы и суммы перевычисляются несколько раз заново; явно вычисляются средние баллы, что приводит к сравнению вещественных чисел; при нахождении трех минимальных значений элементы массива переставляются местами; при печати сравнения производятся с каждым из трех минимальных элементов. Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных</p>	3
<p>Программа работает в целом верно, но выводит только трех худших учеников, даже если кто-то еще сдал экзамены так же. Возможно, в реализации алгоритма содержатся 1–2 ошибки (используется знак “<” вместо “>”, “or” вместо “and” и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие до пяти синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных</p>	2
<p>Программа неверно работает при некоторых входных данных и, возможно, содержит ошибку в алгоритме поиска трех минимальных элементов. Допускается до 4 различных ошибок в ходе решения задачи, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, не допустимая для соответствующего типа данных</p>	1
Задание выполнено неверно	0
Максимальный балл	4

ПРИЛОЖЕНИЯ

Приложение 1

Статистика выполнения заданий экзаменационной работы по информатике

№ п/п	Обозначение задания в работе	Проверяемые элементы содержания и виды деятельности	Средний % вы- полнения
1	A1	Кодирование текстовой информации. Кодировка ASCII. Основные используемые кодировки кириллицы	88
2	A2	Знания о методах измерения количества информации	69
3	A3	Умение подсчитывать информационный объем сообщения	48
4	A4	Знания о системах счисления и двоичном представлении информации в памяти компьютера	80
5	A5	Умение выполнять арифметические операции в двоичной, восьмеричной и шестнадцатеричной системах счисления	71
6	A6	Знание и умение использовать основные алгоритмические конструкции: следование, ветвление, цикл	77
7	A7	Использование переменных. Объявление переменной (тип, имя, значение). Локальные и глобальные переменные	81
8	A8	Работа с массивами (заполнение, считывание, поиск, сортировка, массовые операции и пр.)	66
9	A9	Знание основных понятий и законов математической логики	69
10	A10	Умение строить и преобразовывать логические выражения	73
11	A11	Умение строить таблицы истинности и логические схемы	80
12	A12	Умение представлять и считывать данные в различных типах информационных моделей (схемы, карты, таблицы, графики и формулы).	87
13	A13	Умение кодировать и декодировать информацию	77
14	A14	Формальное исполнение алгоритма, записанного на естественном языке	86
15	A15	Знания о файловой системе организации данных	85

№ п/п	Обозначение задания в работе	Проверяемые элементы содержания и виды деятельности	Средний % вы- полнения
16	A16	Знание технологии хранения, поиска и сортировки информации в базах данных	62
17	A17	Знание технологии обработки графической информации	93
18	A18	Знание технологии обработки информации в электронных таблицах	76
19	A19	Знания о визуализации данных с помощью диаграмм и графиков	74
20	A20	Умение исполнить алгоритм, записанный в виде блок-схемы или программы на алгоритмическом языке	72
21	B1	Представление числовой информации в памяти компьютера. Перевод, сложение и умножение в разных системах счисления	54
22	B2	Умение строить и преобразовывать логические выражения	67
23	B3	Умение исполнять алгоритм в среде формального исполнителя	87
24	B4	Умение строить и преобразовывать логические выражения	56
25	B5	Умение определять скорость передачи информации при заданной пропускной способности канала	46
26	B6	Умение исполнять алгоритм, записанный на естественном языке	55
27	B7	Знание базовых принципов организации и функционирования компьютерных сетей, адресации в Сети	88
28	B8	Умение осуществлять поиск информации в Интернете	61
29	C1	Умение прочесть фрагмент программы на языке программирования и исправить допущенные ошибки	48
30	C2	Умение написать короткую (10—15 строк) простую программу обработки массива на языке программирования или записать алгоритм на естественном языке	40
31	C3	Умение построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию	41
32	C4	Умение создавать собственные программы (30—50 строк) для решения задач средней сложности	12

Предмет	“2”						“3”					
	2005		2006		2007		2005		2006		2007	
	Интервал баллов/доля участников, %						Интервал баллов/доля участников, %					
	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых
1	2	3	4	5	6	7	8	9	10	11	12	13
Рус- ский язык	0-17	0-30	0-18	0-30	0-18	0-29	18- 35	31- 50	19- 35	31- 49	19- 35	30- 48
	8,00		9,10		9,31		46,13		43,30		39,90	
Мате- матика	0-5		0-5		0-6		6-11		6-11		7-12	30- 48
	22,91		21,32		23,72		35,07		34,18		35,06	
Физика	0-10	0-33	0-9	0-34	0-11	0-32	11- 20	34- 50	10- 19	35- 51	12- 22	33- 51
	14,84		17,66		18,99		44,46		44,87		44,27	
Химия	0-15	0-31	0-15	0-30	16- 33	32- 49	16- 34	31- 49	17- 35	31- 49	17- 35	31- 49
	21,27		22,60		23,40		40,48		37,78		36,53	
Инфор- матика			0-11	0-25	0-11	0-24			12- 22	26- 46	12- 22	25- 46
			19,33		15,27				39,27		35,11	
Биоло- гия	0-16	0-31	0-16	0-31	0-17	0-28	17- 33	32- 48	17- 34	32- 49	18- 36	29- 48
	7,81		9,66		11,85		47,15		50,09		46,83	
Исто- рия	0-15	0-32	0-16	0-32	0-15	0-32	16- 32	33- 49	17- 33	33- 49	16- 32	33- 50
	15,70		16,60		16,72		43,85		42,33		42,05	
Геогра- фия	0-15	0-36	0-35	0-16	0-16	0-32	16- 29	37- 50	16- 30	36- 51	17- 31	33- 49
	16,31		16,76		17,58		44,94		43,52		41,07	
Ан- глий- ский язык		0-35	0-30	0-30	0-30	0-30		36- 63	31- 58	31- 58	31- 58	31- 58
	15,72		16,09		16,46		33,28		28,60		27,89	
Не- мецкий язык		0-35	0-30	0-30	0-30	0-30		36- 63	31- 58	31- 58	31- 58	31- 58
	22,76		23,36		21,38		32,67		33,90		35,64	
Фран- цузский язык		0-35	0-30	0-30	0-30	0-30		36- 63	31- 58	31- 58	31- 58	31- 58
	16,02		10,67		9,79		26,54		27,38		23,96	
Обще- ство- вание	0-19	0-30	0-18	0-33	0-19	0-32	20- 33	31- 44	19- 32	34- 47	20- 33	33- 47
	6,38		7,91		7,98		32,02		33,96		33,34	
Лите- ратура	0-13	0-33	0-13	0-36	0-11	0-36	14- 27	34- 50	14- 27	37- 51	12- 26	37- 55
	16,09		18,12		19,23		39,80		40,88		43,40	

Приложение 2

“4”						“5”					
2005		2006		2007		2005		2006		2007	
Интервал баллов/доля участников, %						Интервал баллов/доля участников, %					
пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых	пер- вич- ных	тесто- вых
14	15	16	17	18	19	20	21	22	23	24	25
36-48	51-68	36-48	50-66	36-49	49-59	49-59	69-100	49-60	67-100	50-60	66-100
35,93		36,12		38,31		9,93		11,48		12,48	
12-18		12-18		13-18		19-30		19-30		19-30	
31,39		33,14		33,14		31,88		10,64		11,35	9,34
21-35	51-67	20-33	52-69	23-35	52-68	36-52	68-100	34-52	70-100	36-52	69-100
32,53		27,88		27,13		8,17		9,59		9,61	
34-50	50-63	35-51	50-66	36-52	50-66	51-65	64-100	52-67	67-100	53-67	67-100
27,30		25,70		26,63		10,94		13,92		13,45	
		23-32	47-68	-23-1	47-65			33-40	69-100	32-40	66-100
		31,76		35,83				9,64		13,79	
34-50	49-65	35-51	51-66	37-53	49-66	51-66	66-100	52-69	67-100	54-69	67-100
34,84		30,19		29,85		10,20		10,06		11,47	
33-48	50-65	34-49	50-65	33-49	51-67	49-9	66-100	50-69	66-100	50-68	68-100
28,28		28,21		29,13		12,17		12,85		12,09	
30-43	51-65	31-45	52-67	32-47	50-67	44-58	66-100	46-60	68-100	48-60	68-100
30,14		30,07		31,73		8,60		9,64		9,62	
	64-88	59-83	59-83	59-83	59-83		89-100	84-100	84-100	84-100	84-100
42,41		42,60		41,29		8,58		12,71		14,35	
	64-88	59-83	59-83	59-83	59-83		89-100	84-100	84-100	84-100	84-100
29,91		27,09		30,57		14,66		15,65		12,42	
	64-88	59-83	59-83	59-83	59-83		89-100	84-100	84-100	84-100	84-100
40,96		41,07		47,29		16,48		20,88		18,96	
34-46	45-58	33-44	48-60	34-46	48-61	47-64	59-100	45-62	61-100	47-62	62-100
41,59		37,90		41,22		20,01		20,23		17,47	
28-40	51-66	28-40	52-66	27-35	56-66	41-51	67-100	41-51	67-100	36-40	67-100
34,72		31,79		24,33		9,39		3,21		13,04	

СОДЕРЖАНИЕ

<i>Предисловие</i>	3
<i>Введение</i>	4

ЭКЗАМЕНАЦИОННЫЕ ЗАДАНИЯ

Раздел 1. Информационные процессы и системы	10
Информация и ее кодирование	10
Алгоритмизация и программирование	16
Основы логики	35
Моделирование	41
Раздел 2. Информационные и коммуникационные технологии	44
Программные средства информационных и коммуникационных технологий	44
Обработка графической информации	46
Обработка информации в электронных таблицах	47
Хранение, поиск и сортировка информации в базах данных ..	53
Телекоммуникационные технологии	56
Технологии программирования	58

КОММЕНТАРИИ

Раздел 1. Информационные процессы и системы	68
Информация и ее кодирование	68
Алгоритмизация и программирование	74
Основы логики	89
Моделирование	94
Раздел 2. Информационные процессы и системы	97
Общее замечание	97
Программные средства информационных и коммуникационных технологий	97
Обработка графической информации	98
Обработка информации в электронных таблицах	99
Хранение, поиск и сортировка информации в базах данных ..	103
Телекоммуникационные технологии	104
Технологии программирования	106
Ответы	112
Приложения	154
<i>Приложение 1. Статистика выполнения заданий экзаменационной работы по информатике</i>	154
<i>Приложение 2. Сравнение результатов ЕГЭ</i>	157

ФЕДЕРАЛЬНЫЙ БАНК ЭКЗАМЕНАЦИОННЫХ МАТЕРИАЛОВ

ЕГЭ 2009

ИНФОРМАТИКА

**ФЕДЕРАЛЬНЫЙ БАНК
ЭКЗАМЕНАЦИОННЫХ МАТЕРИАЛОВ**

Авторы-составители:

Якушкин Павел Алексеевич

Крылов Сергей Сергеевич

Директор редакции И Федосова

Ответственный редактор А Жилинская

Редактор Т Судакова

Художественный редактор Е Брынчик

Технический редактор Н Тростянская

Компьютерная верстка А Григорьев

Корректор Н Друх

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**

Подписано в печать 12.02.2009

Формат 60х90^{1/16} Гарнитура «Школьная» Печать офсетная

Бумага газ Усл. печ. л. 10,0

Тираж 3000 экз. Заказ № 139

Отпечатано с готовых диапозитивов

в ГП ПО «Псковская областная типография»

180004, г. Псков, ул. Ротная, 34