



ФЕДЕРАЛЬНЫЙ ИНСТИТУТ  
ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ

**ЕГЭ**

**ЕДИНЫЙ ГОСУДАРСТВЕННЫЙ ЭКЗАМЕН**  
Универсальные материалы  
для подготовки учащихся

# ИНФОРМАТИКА

2011

2010

2009





**ФИПИ**

**ЕДИНЫЙ  
ГОСУДАРСТВЕННЫЙ  
ЭКЗАМЕН  
2009**

# **ИНФОРМАТИКА**

**УНИВЕРСАЛЬНЫЕ  
МАТЕРИАЛЫ ДЛЯ ПОДГОТОВКИ  
УЧАЩИХСЯ**

**Под редакцией В.Р. Лешинера**



**«Интеллект-Центр»  
2009**

УДК 373.167.1:004

ББК 32.81я721

Е33



**Авторы-составители**

**Крылов Сергей Сергеевич, Лешинер Вячеслав Роевлович,**

**Якушкин Павел Алексеевич**

**Е33** Единый государственный экзамен 2009. Информатика. Универсальные материалы для подготовки учащихся / Под редакцией В.Р. Лешинера / ФИПИ. – М.: Интеллект-Центр, 2009. – 136 с.



Пособие предназначено для подготовки к сдаче единого государственного экзамена по информатике 2009 г. Содержит общие рекомендации по подготовке к экзамену, тренировочные примеры по каждому типу заданий, тренировочные варианты экзамена с решениями и ответами. Рассмотрены наиболее важные, с точки зрения подготовки к экзамену, теоретические вопросы и даны практические рекомендации по выполнению заданий.

**Генеральный директор издательства «Интеллект-Центр»**

**Миндюк М.Б.**

Подписано в печать 24.10.2008 г. Формат 60х90 1/8.

Печать офсетная. Усл. печ. л. 17,0. Тираж 10000 экз.

Заказ 2498. Отпечатано в ОАО «Щербинская типография»

117623, г. Москва, ул. Типографская, д. 10. Тел. 659-23-27

ISBN 978-5-89790-537-9

© ФИПИ, 2009

© «Интеллект-Центр», 2009

© Художественное оформление  
«Интеллект-Центр», 2009

Пособия серии «Единый государственный экзамен Универсальные материалы для подготовки учащихся» адресованы в первую очередь учителям и выпускникам 2009 года, а также всем интересующимся новой формой государственной (итоговой) аттестации. Основная цель пособия – предоставить учителям и учащимся достоверную информацию о содержании единого государственного экзамена (ЕГЭ). И учителя, и будущие участники экзамена найдут в пособии реальные тестовые задания, которые были представлены на ЕГЭ в предыдущие годы.

Учителя выпускных классов смогут использовать материалы данного пособия в 10–11 классах, как на этапе обучения, так и для организации повторения, осуществления текущего и итогового контроля. Тестовые задания, конкретизирующие требования к подготовке выпускников средней школы по общеобразовательному предмету, сгруппированы в пособии по основным темам школьного курса. В каждом тематическом блоке представлены задания различного уровня сложности, что может помочь учителю организовать дифференцированную подготовку к экзамену.

Выпускникам материалы пособия позволят самостоятельно проверить и оценить уровень своей подготовки в любой период учебного года. Работая с тренировочными вариантами, будущие участники экзамена смогут выработать индивидуальную стратегию выполнения экзаменационной работы.

Кроме того, читатели найдут в книге анализ результатов ЕГЭ 2008 года и рекомендации, которые могут помочь выпускникам при подготовке к экзамену 2009 года.

При подготовке пособия был использован открытый сегмент Федерального банка экзаменационных материалов, размещенный на сайте Федерального института педагогических измерений ([www.fipi.ru](http://www.fipi.ru)) – официального разработчика тестовых заданий ЕГЭ.

**Желаю успеха!**

*Директор Федерального института  
педагогических измерений А.Г. Ершов*

## АНАЛИЗ РЕЗУЛЬТАТОВ ЭКЗАМЕНА 2008 ГОДА ПО ИНФОРМАТИКЕ

Назначение Единого государственного экзамена по информатике — оценить общеобразовательную подготовку по информатике выпускников XI (XII) классов общеобразовательных учреждений и абитуриентов с целью итоговой государственной аттестации учащихся и отбора для зачисления в учреждения высшего профессионального образования. В 2008г. число участников экзамена превысило 10 тыс. человек, что значительно больше, чем в предыдущем году.

**Число участников экзамена**

Пол	Число экзаменуемых		Процент экзаменуемых	
	ЕГЭ 2007	ЕГЭ 2008	ЕГЭ 2007	ЕГЭ 2008
Девушки	600	2973	22,3	28,7
Юноши	2094	7374	77,7	71,3
Все	2694	10347	100	100

Содержание экзаменационной работы 2008г. определялось на основе утвержденного Министерством образования Российской Федерации обязательного минимума содержания среднего (полного) общего образования по информатике (Приказ от 30.06.99 №56) с учетом тенденций развития предмета, заложенных в образовательном стандарте 2004 г.

Экзаменационная работа по структуре не отличалась от работы 2007 г. и состояла из трех частей.

Часть 1 содержала 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 включала задания по темам: "Информация и её кодирование", "Основы логики", "Алгоритмизация и программирование", "Телекоммуникационные технологии" — всего 8 заданий с кратким ответом.

Задания Части 3 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 записывались в развернутой форме и проверялись экспертами региональных предметных комиссий. За выполнение каждого задания давалось определенное количество баллов, в зависимости от полноты и качества выполнения, поэтому задания 3 части при общем количестве заданий 4 (12,5% общего количества заданий) давали при максимальном выполнении 12 первичных баллов, что составляет 30% общего количества первичных баллов. С другой стороны, эти задания были самыми сложными и самыми трудоемкими, рекомендованное время их выполнения в два раза превосходило время, отводимое на выполнение первых двух частей работы.

Содержание экзамена включает основные темы курса информатики и информационных технологий, объединенные в следующие тематические блоки: "Информация и её 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".

В целом работа 2008 г. по структуре и трудности соответствовала работе 2007 г. Экзаменационная работа 2009 г. в очень большой степени будет соответствовать работе 2008 в части структуры и сравнительной сложности заданий.

Правильный ответ на каждое задание первой и второй части приносит учащемуся один балл. Таким образом, правильные ответы на все задания первой и второй части дают 28 тестовых баллов из максимальных 40 баллов (70%). Остальные 12 тестовых баллов экзаменующийся может набрать, выполняя четыре задания третьей части работы. Набранное количество тестовых баллов по специальной формуле переводится в итоговый балл по 100-балльной шкале, который и выставляется в сертификат ЕГЭ. Очевидно, что задания первой и второй части экзамена имеют разную сложность, но правильный ответ на каждое из них оценивается одним тестовым баллом. Поэтому шкала перевода тестовых баллов в итоговые – нелинейная, на результат, полученный абитуриентом на экзамене, влияет выполнение работы всеми остальными участниками тестирования.

Опыт Единого Государственного экзамена 2007 и 2008 гг. показал, что большинство учащихся, выбравших информатику в качестве экзаменационного предмета, освоили основное содержание предмета, определяемое нормативными документами, но при подготовке учащихся к ЕГЭ необходимо также обращать их внимание на темы, традиционно интересующие высшие учебные заведения: алгоритмизацию и программирование. Учащиеся для успешной сдачи экзамена должны не только знать основные алгоритмические конструкции и операторы изучаемого языка программирования, но и иметь опыт самостоятельной записи алгоритмов и программ, решения практических задач методом разработки и отладки компьютерной программы. Следует уделять больше внимания формализации записи и исполнения алгоритмов, так как результаты экзамена показывают, что у части учащихся так и не формируется умение формального исполнения алгоритмов.

Рассмотрим результаты выполнения заданий ЕГЭ по информатике 2008 г. по основным разделам курса.

### **Информация и ее кодирование**

Проверке знаний и умений по этому разделу содержания курса информатики посвящено 8 заданий, из которых шесть – с выбором ответа и два – с кратким ответом. Средний процент выполнения колеблется от 75% (задание А4, знание о системах счисления и двоичном представлении информации в памяти компьютера) до 45% (задание А3, умение подсчитывать информационный объем сообщения). Помимо задания А4, не вызвали затруднений задания А1 и А13 на двоичное кодирование – средний процент выполнения от 63% до 83%. Сложным оказалось задание В1 на знание математических основ записи чисел в позиционных системах счисления (57% выполнения). Рост показателей выполнения заданий этой темы от года к году наглядно демонстрирует справедливость тезиса о том, что публикация демонстрационных версий и вариантов экзаменационных работ прошлых лет способствует привлечению внимания учителей и экзаменуемых к определенным типам заданий. Так как данная тема хорошо изложена в учебниках и имеет устоявшееся содержание, большинство абитуриентов показывает по этой теме удовлетворительные результаты.

### **Алгоритмизация и программирование**

Этот раздел курса был представлен в экзаменационной работе наиболее подробно: в общей сложности 9 заданий базового, повышенного и высокого уровней сложности во всех трех разделах работы. 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 ифры.

Экзаменующиеся справились с заданием А6 на анализ и исполнение алгоритма, записанного в виде блок-схемы. (Средний процент 77%). Задание В3 на запись фрагмента алгоритма для исполнителя с фиксированным набором команд выполнили в среднем 72% экзаменующихся. Задание А7 на использование переменных также не вызвало затруднений – средний процент выполнения в 2008 г. составил 76%. Задание повышенного уровня

А8, проверяющее знание алгоритмов работы с массивами, в 2008 г правильно выполнили всего 51% экзаменовавшихся. Традиционно хорошо выполняется задание А14 базового уровня сложности, проверяющее умение исполнить алгоритм, записанный на естественном языке. Средний процент его выполнения в 2008 г составил 84% при 83% в 2007 г. Задание А20 на проверку умения формально исполнить алгоритм, было отнесено к типу заданий на применение знаний в новой ситуации, с ним справились всего 26% участников.

Следует отметить, что при том, что в формулировке задания С2 традиционно значится «запишите на русском языке или языке программирования», большинство экзаменуемых записывает ответ на известном им языке программирования. Это неудивительно, так как задание формально записать алгоритм на естественном языке является для школьников относительно новым. В современной российской школе уделяется недостаточно внимания (не только на информатике) обучению школьников умению грамотно и четко формулировать целостные логические (алгоритмические) высказывания на родном языке, в то время как упражнения на составление простых программ обработки массивов предполагаются большинством школьных программ по информатике.

В целом показано хорошее знание темы абитуриентами, что объясняется ее центральным положением в школьном курсе информатики и хорошо отработанным за долгие годы развития предмета содержанием обучения. В 2008 г. также стал заметен результат целенаправленной подготовки школьников к ЕГЭ по информатике с использованием опубликованных вариантов и учебно-тренировочных материалов.

### **Основы логики**

По данному разделу в экзаменационной работе содержалось пять заданий: три с выбором ответа и два с кратким ответом. Учащиеся хорошо справились с заданием А11 базового уровня на проверку умения строить таблицы истинности и логические схемы: 86% выполнения в среднем (78% выполнения в 2007 г.), а также с заданием А10 базового уровня на преобразование логических выражений. 86% выполнения в среднем при 79% в 2007 г. Результат выполнения задания А9 на проверку знания основных понятий и законов математической логики – 71% (при 57% в 2007 г.).

Задание В2 на решение логического уравнения выполнило в среднем 48% участников. В целом в 2008 году закрепились устойчивая тенденция роста результатов выполнения по теме «основы логики», что, видимо, связано с тем, какое большое внимание было уделено этому разделу при разборе результатов ЕГЭ предыдущих лет. На сегодняшний день результаты выполнения абитуриентами заданий по этой теме вполне соответствуют результатам по первым двум темам.

### **Моделирование**

По теме «моделирование» в экзамене 2008 г. было только одно задание базового уровня с выбором ответа, с которым справились в среднем 70% учащихся.

### **Информационные технологии**

Разделу «Основы информационных технологий» было посвящено 7 заданий в первой и второй частях работы. Анализ этой части работы показывает, что учащиеся имеют хорошее представление о файловой системе организации данных. (Задание А15 базового уровня на воспроизведение знаний, среднее выполнение 93% в 2008 г., 82% в 2007г.) Вызвало некоторые затруднения задание В7 с кратким ответом на адресацию в сетях – 48% выполнения.

Задание В8 повышенного уровня из раздела «Телекоммуникационные технологии» на прогнозирование результатов поиска информации в Интернете в 2008 г. дало результат 62%, что выше результата, показанного в 2007 г. (55%). Этот результат подтверждают совершенствование практики использования компьютеров и Интернет современной молодежью. То задание, которое в 2005 г. казались многим умозрительными и чисто теорети-

ческими, сейчас в большинстве случаев выполняются экзаменующимися с опорой на жизненный опыт.

51% выполнения у задания повышенного уровня А17 из раздела «Технология обработки графической информации».

Другие задания данного раздела были выполнены экзаменующимися очень ровно. Наиболее простым оказалось задание А19 базового уровня на чтение данных, представленных в виде диаграмм – процент выполнения 83% (2007 г. – 88%). Остальные задания были выполнены с похожими результатами: 68% выполнения у заданий А16 (базы данных, в 2007 г. – 71%) и А18 (электронные таблицы, результат 2007 г. 71%)

В целом на основании результатов единого государственного экзамена по информатике 2008 г. можно констатировать, что отмечавшиеся ранее пробелы в изучении отдельных тем этого раздела закрыты, уровень знаний участников ЕГЭ по информатике соответствует требованиям государственного стандарта образования.

### **Программирование**

Два задания с развернутым ответом (С1 и С4) были отведены для проверки знаний учащихся по технологии программирования. Одно задание повышенного уровня сложности предполагало поиск и устранение ошибок в уже имеющейся программе и ее доработку, другое предполагало самостоятельное написание программы для решения оригинальной задачи (высокий уровень сложности). Задачи именно этого раздела информатики традиционно являются одними из самых важных при определении уровня подготовки выпускников в вузах, где практикуют вступительные испытания по предмету. И входя в состав КИМ, эти два задания вносят существенный вклад, позволяя экзаменуемому набрать до 7 баллов (из 40 первичных) в результате их успешного решения.

В данной книге приведены задания, сгруппированные в соответствии с кодификатором содержания курса информатики, и два равноценных тренировочных варианта, при составлении которых использован открытый сегмент федерального банка тренировочных заданий. Все задания снабжены ответами, решения типовых заданий подробно разобраны в разделе «Методические рекомендации по подготовке к ЕГЭ 2009». Там же рассмотрены наиболее важные, с точки зрения подготовки к экзамену, теоретические вопросы и даны практические рекомендации по выполнению заданий.



# МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЕГЭ 2009

## Общие замечания

ЕГЭ 2009 года по информатике сохраняет преемственность по отношению к ЕГЭ 2006-2008 гг., поэтому для подготовки к нему могут быть использованы учебно-тренировочные материалы "ЕГЭ Информатика" (Крылов С.С., Лешинер В.Р., Супрун П.Г., Якушкин П.А.; Под ред. Лешинера В.Р. ~ М. Интеллект-Центр, 2006, 2007, 2008. Часть приведенных ниже примеров взята из этих книг

Практика последних лет показывает, что источником большого количества ошибок является слабая математическая подготовка выпускников. Значительное количество заданий оказываются выполненными неправильно только из-за арифметических ошибок. Учащиеся плохо знают таблицу значений функции  $2^n$  для первых 10 аргументов. При расчете задач на пропускную способность канала связи учащиеся не пользуются двоичными логарифмами, а вместо этого перемножают числа «в столбик», делая при этом ошибки. Часто ошибки допускаются при нахождении остатка при делении целых чисел, при построении углов в 30, 60, 120 градусов и так далее. Притом, что все вышеперечисленные умения должны формироваться при изучении математики в основной школе, их отсутствие сказывается на результатах экзамена по информатике за курс полной средней школы в силу высокой роли межпредметных связей в курсе информатики.

Существенным источником ошибок также является невнимательность учащихся при заполнении бланков второй части работы. 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. При подготовке учащихся к экзамену надо 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 проиллюстрировать тем самым необходимость записывать ответ с помощью букв и цифр стандартной формы, максимально соответствующих образцу, приведенному на бланке для записи ответов.

Готовя учащихся к экзамену, следует 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. В ином случае вероятность ошибок при оценивании работы резко возрастает.

Хорошим подспорьем для изучения многих тем и разделов предмета учащимися, освоения ими необходимых навыков в практической деятельности, приведения в целостную систему мог бы стать организованный комплекс межпредметных проектов задействующих новые информационные технологии в качестве реального инструмента для решения предметных задач.

# 1. Информационные процессы и системы

## 1.1. Информация и ее кодирование

### *Единицы измерения и методы измерения количества информации. Скорость передачи информации*

Напомним основные положения алфавитного подхода к измерению количества информации.

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

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

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

#### **Пример:**

В зрительном зале две прямоугольные области зрительских кресел: одна 10 на 5, а другая 4 на 8. Какое минимальное количество бит потребуется для кодирования каждого места в автоматизированной системе?

#### **Решение.**

$$N = 10 \times 5 + 4 \times 8 = 82.$$

$$N' = 128 = 2^7$$

$$I = \log_2 N' = \log_2(2^7) = 7$$

**Ответ:** 7 бит.

#### **Пример:**

В велоспорте участвуют 230 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 20 велосипедистов?

#### **Решение:**

$$N = 230$$

$$N' = 256 = 2^8$$

$$I = \log_2 N' = 8 - \text{это минимальное количество бит для записи номера спортсмена.}$$

Поскольку была записана информация о 20 спортсменах, объем записанного сообщения составил  $20 \times 8 = 160$  бит.

**Ответ:** 160 бит

#### **Пример:**

Сколько существует различных последовательностей из символов «+» и «-», длиной ровно в шесть символов?

#### **Решение.**

В данном случае алфавит состоит из двух элементов, потому информационный объем одного символа – 1 бит. 6 бит позволяют закодировать множество из  $2^6 = 64$  элементов.

**Ответ:** 64.

При решении этого примера возможен также комбинаторный подход.

**Пример:**

Световое табло состоит из лампочек. Каждая лампочка может находиться в одном из трех состояний («включено», «выключено» или «мигает»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 18 различных сигналов?

**Решение:**

С помощью  $n$  лампочек, каждая из которых может находиться в трех состояниях, можно закодировать  $3^n$  сигналов.  $3^2 < 18 < 3^3$ , поэтому двух лампочек недостаточно, а трех хватит.

**Ответ:** 3

**Пример:**

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 480 бит. Какова длина сообщения в символах?

**Решение:**

После перекодировки из 16-битного кода в 8-битный каждый символ сообщения стал занимать на 8 бит меньше, а все сообщение уменьшилось на 480 бит, следовательно сообщение состояло из  $480 : 8 = 60$  символов.

**Представление числовой информации. Системы счисления.**

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

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

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

Так, в числе  $1987_{10}$  цифра 1 обозначает одну тысячу (третья степень), 9 – девять сотен (вторая степень), 8 – восемь десятков (первая степень) и 7 – семь единиц (нулевая степень). Т. е.  $1987_{10} = 1 \cdot 10^3 + 9 \cdot 10^2 + 8 \cdot 10^1 + 7 \cdot 10^0 = 1000 + 900 + 80 + 7$ . Аналогично, число  $1010_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8 + 2 = 10_{10}$ .

**Пример:**

В системе счисления с некоторым основанием число 12 записывается в виде 110. Укажите это основание.

**Решение:**

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

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

**Ответ:** 3.

**Пример:**

Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 17 оканчивается на 2.

**Решение:**

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

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

**Ответ:** 3, 5, 15.

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

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

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

$$\begin{array}{cccccccc}
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
 128 & & 32 & & 8 & & 2 & 
 \end{array}$$

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

Для перевода из десятичной в двоичную систему надо разложить данное число на степени двойки методом вычитания старшей степени. Например,

$$170 = 128 + 42 = 128 + 0 \cdot 64 + 1 \cdot 32 + 10 = 128 + 0 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 2 =$$

$$= 128 + 0 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$$

Выписав двоичные цифры, получаем равенство  $170_{10} = 10101010_2$ .

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

**Пример:**

Как представляется число  $25_{10}$  в двоичной системе счисления?

- 1)  $1001_2$                       2)  $11001_2$                       3)  $10011_2$                       4)  $11010_2$

**Решение:**

Разложим 25 на степени двойки, начиная со старшей (16).

$$25 = 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 11001_2.$$

Другой способ решения – методом подстановки ответов – можно применить, если вы забыли алгоритм перевода из десятичной в двоичную систему, но помните обратный:

- 1)  $1001_2 = 8 + 0 + 0 + 1 = 9_{10};$                       2)  $11001_2 = 16 + 8 + 0 + 0 + 1 = 25_{10};$   
 3)  $10011_2 = 16 + 0 + 0 + 2 + 1 = 19_{10};$                       4)  $11010_2 = 16 + 8 + 0 + 2 + 0 = 26_{10}.$

**Ответ:** 2.

Арифметические операции в позиционных системах счисления производятся по единому алгоритму. Так, сложение двоичных чисел происходит по классическому алгоритму «столбиком» с переносом двойки в следующий разряд

Покажем этот алгоритм на примере двух двоичных чисел  $1010101_2$  и  $110111_2$ :

Дописываемые единицы	1	1	1		1	1	1	
Первое слагаемое		1	0	1	0	1	0	1
Второе слагаемое		0	1	1	0	1	1	1
Сумма	1	0	0	0	1	1	0	0

Результат сложения выглядит как  $10001100_2$ . Проверим результат нашего сложения, для чего переведем все числа в десятичную систему:  $1010101_2 = 85_{10}$ ,  $110111_2 = 55_{10}$ ,  $10001100_2 = 140_{10}$ , что действительно представляет собой результат сложения 55 и 85.

Двоичная система, являющаяся основой всей компьютерной арифметики, весьма громоздка и неудобна для использования человеком. Поэтому программисты пользуются двумя кратными двоичной системами счисления восьмеричной и шестнадцатеричной. В случае шестнадцатеричной системы арабских цифр не хватает, и в качестве цифр используются первые шесть заглавных букв латинского алфавита. Приведем в качестве примера запись натуральных чисел от единицы до шестнадцати в четырех системах счисления:

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

Видно, что в двоичной системе запись чисел второй восьмерки чисел (от 8 до 15) отличается от записи первой восьмерки (от 0 до 7) наличием единицы в четвертом (справа) разряде. На этом основан алгоритм перевода двоичных чисел в восьмеричные «по триадам». Для применения этого алгоритма надо разбить двоичное число на тройки цифр (естественно, отсчитывая справа) и записать вместо каждой из троек восьмеричную цифру.

$10101101_2 \rightarrow 10\ 101\ 101 \rightarrow 2\ 5\ 5 \rightarrow 255_8$ .

Крайняя левая тройка может быть неполной (как в примере), это не мешает реализации алгоритма. Для получения полных троек можно приписать слева недостающие значащие нули ( $010 = 10$  независимо от системы счисления).

Убедимся в правильности алгоритма.  $10101101_2 = 128+32+8+4+1 = 173_{10}$ ,

$255_8 = 2 \cdot 64 + 5 \cdot 8 + 5 \cdot 1 = 128+40+5 = 173_{10}$ . Для перевода чисел из восьмеричной системы в двоичную используется обратный алгоритм: восьмеричные цифры заменяются на тройки двоичных цифр (при необходимости слева дописываются недостающие нули):

$$325_8 \rightarrow 3\ 2\ 5 \rightarrow 11\ 010\ 101 \rightarrow 11010101_2$$

В десятичной системе это число будет записано как 213

Для перевода чисел из двоичной системы в шестнадцатеричную используется алгоритм «по тетрадам». Строка двоичных цифр разбивается на четверки и вместо них записываются шестнадцатеричные цифры

$$10101101_2 \rightarrow 1010\ 1101 \rightarrow A\ D \rightarrow AD_{16}.$$

Еще раз проверим правильность вычислений:  $10\ 16 + 13 = 173$ . Аналогично работает и обратный алгоритм: вместо шестнадцатеричных цифр подставляются четверки двоичных цифр.  $213 = 208 + 5 = 13\ 16 + 5$  Увидим, что результат выполнения алгоритма тот же:

$$D5_{16} \rightarrow D\ 5 \rightarrow 1101\ 0101 \rightarrow 11010101_2.$$

*Очень важно не забывать дописывать недостающие нули слева при переводе каждой цифры, чтобы получалась полноценная четверка или тройка двоичных цифр!*

При выполнении заданий важно выбрать одну систему счисления. Лучше всего пользоваться той системой, в которой должен быть представлен результат. Если среди данных чисел десятичных нет, то десятичную систему лучше не использовать. Из восьмеричной системы в шестнадцатеричную (и в обратном направлении) гораздо проще переводить через двоичную систему.

$$\text{Пример: } D5_{16} \rightarrow D\ 5 \rightarrow 1101\ 0101 \rightarrow 11010101_2 \rightarrow 011\ 010\ 101 \rightarrow 325_8$$

### Пример:

Вычислите значение суммы в десятичной системе счисления:

$$10_2 + 10_8 + 10_{16} = ?_{10}$$

1)  $30_{10}$

2)  $26_{10}$

3)  $36_{10}$

4)  $20_{10}$

### Решение:

Переведем все числа в десятичную запись.

$$10_2 + 10_8 + 10_{16} = (1\ 2+0) + (1 \cdot 8+0) + (1 \cdot 16+0) = 2 + 8 + 16 = 26_{10}.$$

Ответ: 2

### Пример:

Вычислите сумму чисел  $x$  и  $y$ , если

$$x = 1110101_2;$$

$$y = 1011011_2.$$

Результат представьте в виде восьмеричного числа

1)  $2110_8$

2)  $298_8$

3)  $320_8$

4)  $318_8$

### Решение:

Существует два способа вычисления искомой суммы. *Первый способ* – сложить числа по правилам сложения двоичных чисел (столбиком, как показано выше), а результат перевести в восьмеричную запись, разбив на триады.

$$1110101_2 + 1011011_2 = 11010000_2 \rightarrow 11\ 010\ 000 \rightarrow 3\ 2\ 0 \rightarrow 320_8.$$

Второй способ состоит в том, чтобы сначала перевести числа в восьмеричную систему, а потом сложить их:

$$1110101_2 \rightarrow 1\ 110\ 101 \rightarrow 1\ 6\ 5 \rightarrow 165_8;$$

$$1011011_2 \rightarrow 1\ 011\ 011 \rightarrow 1\ 3\ 3 \rightarrow 133_8.$$

Теперь столбиком сложим восьмеричные числа  $165_8 + 133_8$ :

Дописываемые единицы	1	1
Первое слагаемое	1	6 5
Второе слагаемое	1	3 3
Сумма	3	2 0

Обратите внимание, что во втором столбике  $6+3+1$  дает десять, что записывается как 12<sub>8</sub>, двойка записывается на свое место, а единица подписывается сверху.

Кстати, среди приведенных вариантов ответа два (вар 2 и 4) содержат недопустимые в восьмеричной системе цифры 8 и 9.

**Ответ:** 3.

### **Пример.**

Для кодирования букв А, Б, В, Г решили использовать двухразрядные последовательные двоичные числа (от 00 до 11, соответственно). Если таким способом закодировать последовательность символов БАВГ и записать результат шестнадцатеричным кодом, то получится

- 1) 4В                      2) 411                      3) BACD                      4) 1023

### **Решение.**

По условию А кодируется как 00, Б – 01, В – 10, Г – 11, поэтому БАВГ будет закодировано как 01001011<sub>2</sub>, что соответствует 4В<sub>16</sub>

## **1.2. Алгоритмизация и программирование**

*Алгоритмы, виды алгоритмов, описания алгоритмов. Формальное исполнение алгоритма.*

При решении заданий этой темы необходимо, прежде всего, 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

Приведем примеры решения типичных задач этого раздела

### **Пример:**

Имеется исполнитель Кузнечик, который живет на числовой оси. Система команд Кузнечика: «Вперед N» (Кузнечик прыгает вперед на N единиц), «Назад M» (Кузнечик прыгает назад на M единиц). Переменные N и M могут принимать любые целые положительные значения. Известно, что Кузнечик выполнил программу из 40 команд, в которой команд «Назад 2» на 10 больше, чем команд «Вперед 3». Других команд в программе не было. На какую одну команду можно заменить эту программу, чтобы Кузнечик оказался в той же точке, что и после выполнения программы?

### **Решение:**

Если всего команд 40, то команд «Назад 2» было 25, а «Вперед 3» всего 15. Кузнечик прыгнул вперед на  $15 \cdot 3 = 45$  шагов, а назад на  $25 \cdot 2 = 50$  шагов. Тем самым, он оказался на 5 шагов назад от первоначальной точки. Последовательность команд в алгоритме в данном случае не имеет значения

**Ответ:** Назад 5.

### **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  $n$** , 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 4 [Команда1 Команда2]** означает, что последовательность команд в скобках повторится 4 раз.

Черепашке был дан для исполнения следующий алгоритм:

**Повтори 4 [Вперед 10 Направо 120]**

Какая фигура появится на экране?

- 1) Незамкнутая ломаная линия
- 2) Правильный треугольник
- 3) Квадрат
- 4) Правильный пятиугольник

**Решение:**

Черепашка прочертит на экране 4 линии, но последний отрезок полностью совпадет с первым, так как после третьего выполнения цикла черепашка полностью обернется вокруг своей оси и окажется в той же точке, что и изначально. Так что на экране появится правильный треугольник.

**Ответ:** 3.

**Пример**

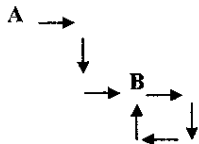
Исполнитель Робот действует на клетчатой доске, между соседними клетками которой могут стоять стены. Робот передвигается по клеткам доски и может выполнять команды 1 (вверх), 2 (вниз), 3 (вправо), 4 (влево), переходя на соседнюю клетку в направлении, указанном в скобках. Если в этом направлении между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

3233241

Какую последовательность из трех 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?

**Решение:**

Начертим траекторию, по которой двигался Робот. Обозначим буквой А – начальную точку его движения, В – конечную. Из рисунка видно, что возвращение из точки В в точку А можно осуществить по программе из трех команд: влево-вверх-влево, т.е. 414.



**Ответ:** 414

**Пример:**

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  $i$ -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,  $a := \text{'дом'}$ ).

Фрагмент алгоритма:

```

i := Длина (a)
k := 1
b := 'П'
пока i > 0
нц
  c := Извлечь (a, i)
  b := 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, записывая значения переменных в таблицу

Выполняемый оператор	a	b	c	i	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
$i := i - k$	'РОЗА'	'ПА'	'А'	3	1
$c := \text{Извлечь}(a, i)$	'РОЗА'	'ПА'	'З'	3	1
$b := \text{Склеить}(b, c)$	'РОЗА'	'ПАЗ'	'З'	3	1
$i := i - k$	'РОЗА'	'ПАЗ'	'З'	2	1
$c := \text{Извлечь}(a, i)$	'РОЗА'	'ПАЗ'	'О'	2	1
$b := \text{Склеить}(b, c)$	'РОЗА'	'ПАЗО'	'О'	2	1
$i := i - k$	'РОЗА'	'ПАЗО'	'О'	1	1
$c := \text{Извлечь}(a, i)$	'РОЗА'	'ПАЗО'	'Р'	1	1
$b := \text{Склеить}(b, c)$	'РОЗА'	'ПАЗОР'	'Р'	1	1
$i := i - k$	'РОЗА'	'ПАЗОР'	'Р'	0	1

Ответ: 2

**Пример:**

Записано 6 строк, каждая имеет свой номер – от 0 до 5.

В нулевой строке записана цифра 0 (ноль).

Каждая последующая строка состоит из двух повторений предыдущей и добавленного в конец своего номера (в  $i$ -й строке в конце приписана цифра  $i$ ). Ниже показаны первые четыре строки, сформированные по описанному правилу (в скобках записан номер строки):

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

Какая цифра стоит в последней строке на 62-м месте (считая слева направо)?

**Решение:**

Найдем длину последней 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:

- (0) 1 элемент в строке;
- (1)  $1 \times 2 + 1 = 3$  элемента в строке;
- (2)  $3 \times 2 + 1 = 7$ ,
- (3)  $7 \times 2 + 1 = 15$  элементов в строке;
- (4)  $15 \times 2 + 1 = 31$ ;
- (5)  $31 \times 2 + 1 = 63$  элемента в строке.

Требуется найти 62-й элемент в строке длиной в 63 символов. Это означает, что нам нужен второй элемент с конца, предпоследний в строке.

Последний символ в последней строке, это ее номер - 5. Предпоследний элемент строки – это последняя цифра в предыдущей строке (по правилу формирования строк). А окончание предыдущей строки – это ее номер, т.е. цифра 4.

**Ответ:** 4

**Пример:**

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 сначала 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) DСВААВААСВААВАА

**Латинский алфавит (для справки):**

АВСDEFGHІJKLМNOPQRSTU VWXYZ

Запишите семь символов подряд, стоящие в восьмой строке со 126-го по 132-е место (считая слева направо).

**Решение:**

Длина  $i$ -й строки составляет  $2^i - 1$  символ, поэтому в 8-й строке со 2-го по 128-й символ и со 129-го по 255-й (включительно) будут находиться все 127 символов 7-й строки. Поэтому символы 126-128 восьмой строки будут совпадать с тремя последними символами 7-й строки, а символы 129-132 8-й строки с четырьмя первыми символами 7-й строки. Три

последних символа любой  $i$ -й строки ( $i > 1$ ) – **В**АА, а  $i$  первых символов  $i$ -й строки не что иное, как первые  $i$  букв латинского алфавита, записанные в обратном порядке, т.е. в нашем случае – **GFEDCBA**. Но нам нужно из них только первые четыре – **GFED**. Осталось только соединить полученные цепочки **В**АА и **GFED**.

Ответ: **В**АА**GFED**

**Замечание.** Формула длины  $i$ -строки не является очевидной. Но при решении этой задачи без неё можно легко обойтись, непосредственно вычислив длину всех строк с 1-й по 8-ю, пользуясь тем, что длина каждой следующей строки равна удвоенной длине предыдущей + 1.

### Пример:

Цепочка из трех бусин формируется по следующему правилу:  
На первом месте в цепочке стоит одна из бусин А, Б, В. На втором – одна из бусин Б, В, Г. На третьем месте – одна из бусин А, В, Г, не стоящая в цепочке на первом или втором месте.

Какая из следующих цепочек создана по этому правилу:

- 1) АГБ                      2) ВАГ                      3) БГГ                      4) ББГ

**Решение.**

#### Полное решение:

На первом месте в цепочке стоит одна из бусин **А, Б, В**.

После выполнения второго условия остаются варианты.

**АБ, АВ, АГ,**

**ББ, БВ, БГ,**

**ВБ, ВВ, ВГ**

На третьем шаге складываются цепочки:

**АБВ, АБГ, АВГ, АГВ,**

**ББА, ББВ, ББГ, БВА, БВГ, БГА, БГВ,**

**ВБА, ВБГ, ВВА, ВВГ, ВГА**

Итого: 16 цепочек, из четырех предложенных подходит только «ББГ».

#### Краткое решение:

В цепочке «АГБ» нарушено правило третьей бусины.

В цепочке «ВАГ» нарушено правило второй бусины.

В цепочке «БГГ» нарушено правило третьей бусины.

В цепочке «ББГ» все правила соблюдаются.

Ответ: 4

### Пример:

Два игрока играют в следующую игру.

Имеется три кучи камней, содержащих соответственно 2, 3, 4 камня. За один ход разрешается или удвоить количество камней в какой-нибудь куче, 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 куче становится  $\geq 15$  камней или во всех трех кучах суммарно становится  $\geq 25$  камней.

Игроки ходят по очереди. Выяснить, кто выигрывает при правильной игре, - первый или второй игрок.

**Решение:**

Для решения задачи составим таблицу (дерево развития игры при различных 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 – 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
2,3,4	4,3,4	8,3,4	проигрыш 1 игрока (при любом продолжении)	
		4,6,4		
		4,3,8		
		6,5,6		
	2,6,4	4,6,4	проигрыш 1 игрока (при любом продолжении)	
		2,12,4		
		2,6,8		
		4,8,6		
	2,3,8		проигрыш 1 игрока (при ходе 2 игрока 2,3,16)	
	4,5,6	8,5,6	выигрыш 1 игрока (при ходе 16,5,6)	
		4,10,6	выигрыш 1 игрока (при ходе 4,20,6)	
		4,5,12	выигрыш 1 игрока (при ходе 4,5,24)	
		6,7,8	выигрыш 1 игрока (при ходе 6,7,16)	

Если 1-й игрок сделает свой первый ход 2,3,4 --> 4,3,4, то 2-й игрок при правильной игре сделает ход 4,3,4 --> 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 --> 2,6,4, то 2-й игрок при правильной игре сделает ход 2,6,4 --> 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 --> 2,3,8, то его проигрыш очевиден, т.к. 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 --> 4,5,6, то он выигрывает игру, т.к. на любой из четырех возможных ответов 2-го игрока (см таблицу) есть выигрывающий ход 1-го игрока.

Таким образом, окончательный **Ответ** к данной задаче:

При правильной игре выигрывает 1-й игрок (при этом его первый ход должен быть 2,3,4 --> 4,5,6).

**Основные алгоритмические конструкции: следование, ветвление, цикл. Блок-схемы. Переменные.**

### Блок-схемы

Блок-схемы являются одним из графических способов представления алгоритмов. Блок-схема алгоритма состоит из блоков, соединенных линиями. Блоки различной формы изображают начало, конец и отдельные шаги алгоритма, а также условие выполнения шага. Выполняемые на каждом шаге действия записываются внутри прямоугольных блоков. Внутри ромбов указываются условия выполнения шагов. Овалами обозначены начало и конец алгоритма. Линии указывают последовательность выполнения действий. Если линия не заканчивается стрелкой, то двигаться по ней нужно в направлении сверху вниз.

### Алгоритмические конструкции

Группа шагов алгоритма, всегда выполняемых последовательно друг за другом без каких-либо условий, называется *линейной последовательностью*.

На рис. 1 изображена линейная последовательность, состоящая из двух шагов.

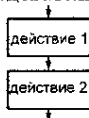


Рис. 1. Линейная последовательность, состоящая из двух шагов

*Ветвление* представляет собой алгоритмическую конструкцию, в которой выполнение того или иного шага зависит от истинности условия. Говорят, что конструкция «ветвление» записана *в полной форме*, если в ней присутствуют команды как для случая истинности условия, так и для случая его ложности.

На рис. 2 приведена блок-схема ветвления в полной форме.

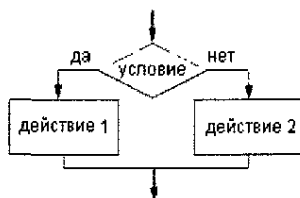


Рис. 2. Блок-схема ветвления в полной форме

Если условие истинно, то будет выполнено только действие 1, в противном случае будет выполнено только действие 2.

Если в ветвлении присутствуют команды только для случая истинности или только для случая ложности условия, то говорят, что она записана *в сокращенной форме*.

На рис. 3 приведены две блок-схемы ветвления в сокращенной форме.

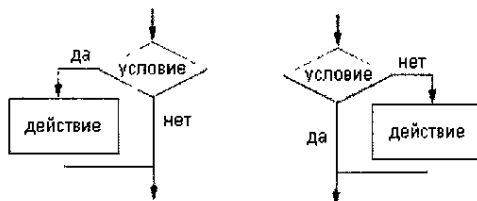


Рис. 3. Блок-схемы ветвления в сокращенной форме.

Цикл представляет собой алгоритмическую конструкцию, в которой многократно выполняется одна и та же последовательность шагов, называемая телом цикла. Каждое однократное исполнение тела цикла называется *итерацией*. Если тело цикла было выполнено N раз, говорят, что было произведено N итераций.

Для того чтобы определить момент прекращения выполнения тела цикла, используется условие цикла. Если при истинности условия цикл продолжается, то такое условие называется условием продолжения цикла. Иными словами, такой цикл продолжается, пока условие цикла истинно.

Если при истинности условия цикл завершается, то такое условие называется условием завершения цикла. В этом случае цикл продолжается до тех пор, пока условие цикла не станет истинным.

Различают циклы с проверкой условия перед выполнением очередной итерации и циклы с проверкой условия после выполнения очередной итерации. Первые называются циклами с предусловием (рис. 5), вторые – с постусловием (рис. 4).

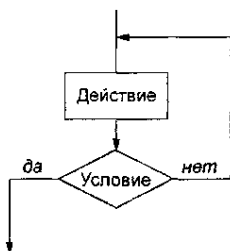


Рис. 4. Блок-схема цикла с постусловием завершения

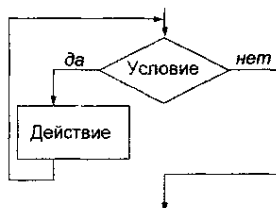


Рис. 5. Блок-схема цикла с предусловием продолжения

Алгоритмическая конструкция называется *вложенной*, если она содержится внутри другой алгоритмической конструкции.

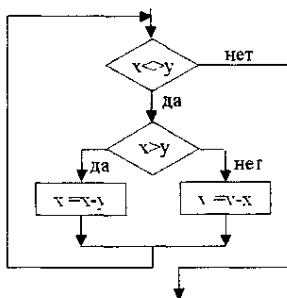
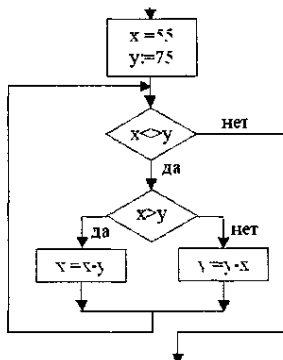


Рис. 6. Команда ветвления вложена в цикл

**Пример:**

Определите значение целочисленной переменной  $x$  после выполнения следующего фрагмента программы:



1) 1

2) 5

3) 10

4) 15

**Решение:**

В блок-схеме присутствует цикл, т.е. одни и те же команды многократно повторяются. Для того, чтобы не ошибиться при выполнении блок-схемы, удобно составить таблицу, в которую заносятся значения переменных и результаты проверки условия на каждом шаге.

**Примечание.** Знак  $\neq$  означает «не равно», знак  $:=$  – присваивание значения.

№ итерации цикла	Значение $x$	Значение $y$	$x \neq y$	$x > y$
0	55	75	55 $\neq$ 75 - да (выполняем тело цикла)	
1	55	75		55 > 75 - нет (вычитаем $x$ из $y$ )
1	55	20		
	55	20	55 $\neq$ 20 - да (выполняем тело цикла)	
2	55	20		55 > 20 - да (вычитаем $y$ из $x$ )
2	35	20		
	35	20	35 $\neq$ 20 - да (выполняем тело цикла)	
3	35	20		35 > 20 - да (вычитаем $y$ из $x$ )
3	15	20		
	15	20	15 $\neq$ 20 - да (выполняем тело цикла)	
4	15	20		35 > 20 - нет (вычитаем $x$ из $y$ )
4	15	5		
	15	5	15 $\neq$ 5 - да (выполняем тело цикла)	

№ итерации цикла	Значение x	Значение y	$x < y$	$x > y$
5	15	5		15 > 5 - да (вычитаем y из x)
5	15	5		
5	10	5		
	10	5	10 < 5 - да (выполняем тело цикла)	
6	10	5		10 > 5 - да (вычитаем y из x)
6	5	5		
	5	5	5 < 5 - нет (алгоритм завершен)	

Итак, переменная x после выполнения данного фрагмента программы приняла значение 5, что соответствует ответу под номером 2 в таблице ответов.

**Ответ. 2.**

**Замечание.** Нетрудно заметить, что блок-схема в этой задаче соответствует известному алгоритму Евклида нахождения наибольшего общего делителя двух чисел (НОД). Поэтому ответ можно получить и без формального выполнения алгоритма, используя, например, такую цепочку умозаключений: «55 делится нацело на 5 (по признаку делимости на 5,  $55 = 5 \cdot 11$ ; 75 нацело на 11 не делится, но тоже делится на 5, следовательно НОД чисел 55 и 75 равен 5».

Если по внешнему виду блок-схемы нелегко понять, какой именно алгоритм она реализует, то для решения подобных задач следует пользоваться общим способом – пошаговым исполнением блок-схемы с заполнением таблицы.

**Исполнение фрагментов программ** В условии задачи приводятся эквивалентные тексты программ на трех алгоритмических языках. Следует выполнять программу на том языке, с которым вы наиболее хорошо знакомы. На остальные два фрагмента не следует обращать внимание, чтобы не терять зря время. Как и в предыдущем задании, для исполнения программы удобно составить таблицу.

**Пример:**

Определите значение целочисленных переменных x, y и t после выполнения фрагмента программы (ниже представлена одна и та же программа, записанная на разных языках программирования):

Бейсик	Паскаль	Алгоритмический
x=5 y=7 t=x ' MOD - стандартная операция, вычисляющая остаток от деления нацело первого аргумента на второй x= y MOD x y=t	x =5; y:=7; t:=x; x:= y mod x; {mod - стандартная операция, вычисляющая остаток от деления нацело первого аргумента на второй} y:=t;	x =5; y:=7, t:=x; x:=mod (y,x);   mod - стандартная функция, вычисляющая остаток от деления нацело первого аргумента на второй  y.=t;

1) x=2, y=5, t=5

2) x=7, y=5, t=5

3) x=2, y=2, t=2

4) x=5, y=5, t=5



**Решение:**

Составим и заполним таблицу

Шаг	Значение $x$ после шага	Значение $y$ после шага	Значение $t$ после шага
$x=5$	5	не определено	не определено
$y=7$	5	7	не определено
$t=x$	5	7	5
$x = y \text{ MOD } x$	2	7	5
$y=t$	2	5	5

Ответ: 1.

**Работа с массивами**

Двумерный массив  $A$  размером  $N \times M$  можно изобразить в виде таблицы, состоящей из  $N$  строк и  $M$  столбцов. Каждый элемент двумерного массива имеет *адрес*, состоящий из имени массива (в нашем случае –  $A$ ) и двух чисел – номера строки и номера столбца. Эти номера называются индексами и записываются через запятую в круглых или прямоугольных скобках. При помощи адреса можно обратиться к нужному элементу массива.

Двумерный массив  $A$  размером  $N \times M$

	1	2	...	$j$	..	$M-1$	$M$
1	$A(1,1)$	$A(1,2)$		$A(1,j)$	..	$A(1,M-1)$	$A(1,M)$
2	$A(2,1)$	$A(2,2)$	...	$A(2,j)$	.	$A(2,M-1)$	$A(2,M)$
...	..	..	..	..	..	..	..
$i$	$A(i,1)$	$A(i,2)$	...	$A(i,j)$	.	$A(i,M-1)$	$A(i,M)$
...	..	..	..	..	..	..	..
$N-1$	$A(N-1,1)$	$A(N-1,2)$	...	$A(N-1,j)$	..	$A(N-1,M-1)$	$A(N-1,M)$
$N$	$A(N,1)$	$A(N,2)$	.	$A(N,j)$	.	$A(N,M-1)$	$A(N,M)$

Для обработки массивов часто используется *цикл со счетчиком* (цикл FOR). Цикл со счетчиком состоит из заголовка и тела цикла. В заголовке указывается начальное и конечное значение счетчика. На каждой итерации значение счетчика автоматически увеличивается. Цикл завершается, когда счетчик достигнет конечного значения. Фактически цикл со счетчиком представляет собой разновидность цикла с предусловием продолжения, заключающимся в том, что значение счетчика находится в заданных границах.

**Пример**

Значения двумерного массива задаются с помощью вложенного оператора цикла в представленном фрагменте программы

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 5 FOR k=1 TO 5 B(n, k)=n+k NEXT k NEXT n	for n:=1 to 5 do for k:=1 to 5 do B[n,k]:=n+k,	нц для n от 1 до 5 нц для k от 1 до 5 B[n, k]=n+k кц кц

Чему будет равно значение  $B(2,4)$ ?

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

**Решение:****1-й способ**

Нарисуем таблицу, соответствующую массиву, и заполним ее значениями в соответствии с программой

$k \backslash n$	1	2	3	4	5
1	2	3	4	5	6
2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9
5	6	7	8	9	0

Элемент  $B(2,4)$  находится на пересечении 2-й строки и четвертого столбца, его значение равно 6, что соответствует четвертому пункту в списке предложенных ответов.

**Ответ: 4**

Заметим, что данное решение является нерациональным, так как нам требовалось найти значение всего лишь одного элемента массива, не зависящее от значений других элементов. Поэтому лучше применить другой способ решения.

**2-й способ.** Из текста программы следует, что значение каждого элемента  $B(i,j)$  вычисляется по формуле  $B(i,j) = i+j$ , следовательно  $B(2,4) = 2+4=6$ .

Способ с заполнением таблицы хоть и универсален, но может оказаться весьма трудоемким, поэтому пользоваться им следует лишь в тех случаях, когда число элементов массива невелико (10-20) или не удастся найти другое решение.

**Пример:**

Все элементы двумерного массива  $A$  размером  $4 \times 4$  элемента первоначально были равны 0. Затем значения элементов меняются с помощью вложенного оператора цикла в представленном фрагменте программы (ниже представлена одна и та же программа, записанная на разных языках программирования)

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 4 FOR k=n TO 4 A(n,k)= 1  NEXT k NEXT n	for n = 1 to 4 do for k:=n to 4 do begin A[n, k]:= 1; end end	нц для n от 1 до 4 нц для k от n до 4 A[n, k]:= 1 кц кц

Сколько элементов массива в результате будут равны 1?

Здесь следует обратить внимание на то, что во вложенном цикле начальное значение  $n$  счетчика  $k$  является переменным, так как значение  $n$  меняется во внешнем цикле. Таким образом, на  $i$ -й итерации внешнего цикла будут обработаны следующие элементы массива.

$A(1,1)$   $A(1,2)$   $A(1,3)$   $A(1,4)$  т.е.  $n=1, k = 1, 2, 3, 4$ ,  
на 2-й итерации.

$A(2,2)$   $A(2,3)$   $A(2,4)$  т.е.  $n=2, k = 2 \dots 4$ ,  
на 3-й итерации:

$A(3,3)$   $A(3,3)$  т.е.  $n=3, k = 3, 4$ ;

на 4-й итерации  
 $A(4,4)$  т.е.  $n=4, k=4$ .

Итак, **ответ:**  $4+3+2+1 = 10$  элементов.

Результат работы программы можно представить в виде таблицы (закрашены обработанные элементы).

$k \backslash n$	1	2	3	4
1				
2				
3				
4				

А если бы обрабатывался массив размером не  $4 \times 4$ , а, скажем,  $1000 \times 1000$  элементов? Решим более общую задачу. Пусть обрабатываемый массив имеет размерность  $M \times M$ , а программа выглядит следующим образом

```
FOR n=1 TO M
  FOR k=n TO M
    A(n, k)=1
  NEXT k
NEXT n
```

(Приведен пример на Бейсике, любители Паскаля или Алгоритмического языка легко смогут сделать аналогичные изменения)

Количества элементов, обрабатываемых на каждой итерации, образуют арифметическую прогрессию из  $M$  членов:  $M, M-1, \dots, 2, 1$ . Сумма такой прогрессии находится по формуле  $M \cdot (M+1)/2$ , что согласуется с полученным нами ранее ответом для случая  $M=4$

### Пример:

Все элементы двумерного массива  $A$  размером  $10 \times 10$  элементов первоначально были равны 1. Затем значения элементов меняются с помощью вложенного оператора цикла в представленном фрагменте программы.

Бейсик	Паскаль	Алгоритмический
<pre>FOR n=1 TO 4   FOR k=1 TO n+1     A(n,k)= A(n,k)-1     A(n,k+1)= A(n,k)-1   NEXT k NEXT n</pre>	<pre>for n:=1 to 4 do   for k:=1 to n+1   begin     A[n, k]:=A[n, k]-1;     A[n, k+1]:=A[n, k]-1;   end</pre>	<pre>нц для n от 1 до 4   нц для k от 1 до n+1     A[n, k]:=A[n, k]-1     A[n, k+1]:=A[n, k]-1   кц</pre>

Сколько элементов массива в результате будут равны 0?

### Решение:

Рассмотрим 1-ю итерацию внешнего цикла, в которой обрабатывается 1-я строка массива

Исходное состояние 1-й строки:

1 1 1 1 1 1 1 1 1 1;

$n=1, k=1$ ;

после шага  $A(n,k) = A(n,k)-1$ , т.е.  $A(1,1) = A(1,1)-1$ , эта строка примет вид

0 1 1 1 1 1 1 1 1;

после шага  $A(n,k+1) = A(n,k)-1$ , т.е.  $A(1,1+1) = A(1,1)-1$ , эта строка примет вид:

0 -1 1 1 1 1 1 1 1;

$n=1, k=2$ ;

после шага  $A(n,k) = A(n,k)-1$ , т.е.  $A(1,2) = A(1,2)-1$  эта строка примет вид:

0 -2 1 1 1 1 1 1 1;

после шага  $A(n,k+1) = A(n,k)-1$ , т.е.  $A(1,3) = A(1,2)-1$  эта строка примет вид:

0 -2 -3 1 1 1 1 1 1;

и т.д.

Таким образом, нулевым станет лишь первый элемент строки, все остальные будут либо отрицательными, либо останутся необработанными (концы строк сохраняют свои единичные значения).

Это справедливо для всех обрабатываемых строк, таких строк 4, следовательно, **Ответ: 4.**

В задании C2 демо-версии 2006г проверяется умение написать короткую (10-15 строк) простую программу обработки массива или записать алгоритм на естественном языке. Для решения этой группы заданий полезно усвоить следующие элементарные алгоритмы.

1) Поиск минимального элемента в массиве

Пусть кандидатом на минимальный элемент является первый элемент массива. Будем просматривать последовательно все оставшиеся элементы массива. Если просматриваемый элемент меньше кандидата на минимальный элемент, то кандидату на минимальный элемент присваивается значение текущего элемента. После завершения просмотра, кандидат в минимальные элементы становится минимальным элементом.<sup>1</sup>

На языке Паскаль это можно записать так:

Var i,minElement: integer; A: array [1..N] of integer;

minElement := A[1];

for i := 2 to N do

if A[i]<minElem then minElem := A[i];

2) Поиск максимального элемента в массиве

Пусть кандидатом на максимальный элемент является первый элемент массива. Будем просматривать последовательно все оставшиеся элементы массива. Если просматриваемый элемент больше кандидата на максимальный элемент, то кандидату на максимальный элемент присваивается значение текущего элемента. После завершения просмотра, кандидат в максимальные элементы становится максимальным элементом

На языке Паскаль это можно записать так:

Var i,maxElement: integer; A: array [1..N] of integer;

...

maxElement := A[1];

for i:= 2 to N do

if A[i]>maxElem then maxElem := A[i];

<sup>1</sup> Интересно, что многие школьники, затрудняющиеся в составлении алгоритма решения задачи “найти разность между максимальным и минимальным элементом массива”, без особых проблем составляют искомый алгоритм, если задачу переформулировать в виде “В витрине выставлен ряд товаров. Двигаясь вдоль витрины, можно просматривать ценники товаров и запоминать, при необходимости, некоторые из них. Найти разность между максимальной и минимальной ценой”

3) Поиск минимального элемента в массиве и его номера. Если минимальных элементов несколько, то будет получен номер первого из них.

Пусть кандидатом на минимальный элемент является первый элемент массива. Его номер равен 1. Будем просматривать последовательно все оставшиеся элементы массива. Если просматриваемый элемент меньше кандидата на минимальный элемент, то кандидату на минимальный элемент присваивается значение текущего элемента, а номеру кандидата на минимальный элемент присваивается номер текущего элемента. После завершения просмотра, кандидат в минимальные элементы становится минимальным элементом, а его номер – номером минимального элемента.

На языке Паскаль это можно записать так:

```
Var i, minElementNumber, minElement: integer; A: array [1..N] of integer;  
...  
minElement := A[1];  
minElementNumber := 1;  
for i := 2 to N do  
    if A[i] < minElem then  
        begin  
            minElem := A[i];  
            minElementNumber := i  
        end;  
end;
```

4) Поиск максимального элемента в массиве и его номера. Этот элементарный алгоритм легко получить из предыдущего, так же, как второй алгоритм получен из первого.

5) Вычисление суммы элементов числового массива.

Для накопления суммы отводится отдельная переменная. Ее значение обнуляется.

Просматриваем последовательно все элементы массива, прибавляя значение каждого к текущему значению этой переменной. После завершения просмотра в ней будет сумма элементов массива.

На Паскале:

```
Var i, summa: integer; A: array [1..N] of integer;  
...  
summa := 0;  
for i := 1 to N do summa := summa + A[i];
```

6) Подсчет числа элементов массива, удовлетворяющих заданному условию:

Для счетчика отводится отдельная переменная. Ее значение обнуляется.

Просматриваем последовательно все элементы массива, проверяя выполнение заданного условия для каждого элемента и прибавляя к значению счетчика единицу, если условие выполняется. После завершения просмотра в переменной-счетчике будет число элементов массива, удовлетворяющих условию.

Пример подсчета отрицательных элементов массива на Паскале:

```
Var i, nNegative: integer; A: array [1..N] of integer;  
...  
nNegative := 0;  
for i := 1 to N do if A[i] < 0 then nNegative := nNegative + 1;
```

После того как алгоритм начерно написан, рекомендуется его протестировать на небольших (3-4 элемента) 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

### 1.3. Основы логики

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

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

Переместительный закон.

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

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

Сочетательный закон:

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

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

Распределительный закон:

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

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

Закон непротиворечия:

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

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

Закон исключенного третьего.

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

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

Закон двойного отрицания:

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

Законы де Моргана:

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

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

В справедливости указанных законов можно убедиться с помощью таблиц истинности.

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

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

**Пример:**

Для какого имени истинно высказывание

$\neg(\text{Первая буква имени гласная} \rightarrow \text{Четвертая буква имени согласная})$

1) ЕЛЕНА	2) ВАДИМ	3) АНТОН	4) ФЕДОР
----------	----------	----------	----------

**Решение (1 способ):**

Поскольку данное высказывание истинно, его отрицание

$(\text{Первая буква имени гласная} \rightarrow \text{Четвертая буква имени согласная})$  – ложно.

Это высказывание является импликацией и ложно только в том случае, когда левая часть его (Первая буква имени гласная) истинна, а вторая (Четвертая буква имени согласная) – ложна. То есть, первая и четвертая буквы имени – гласные

Этому условию удовлетворяет только имя АНТОН.

**Решение (2 способ):**

Воспользуемся тождествами

$$\begin{aligned} a \rightarrow b &= (\neg a) \vee b \\ \neg(a \vee b) &= (\neg a) \wedge (\neg b) \\ \neg(\neg a) &= a \end{aligned}$$

Используя эти тождества, получим высказывание в виде

$$\neg(a \rightarrow b) = \neg(\neg a \vee b) = a \wedge (\neg b),$$

где  $a$  = «Первая буква имени гласная»,

$b$  = «Четвертая буква имени согласная», т.е. получим высказывание: Первая буква имени гласная  $\wedge \neg$ (Четвертая буква имени согласная).

Или:

Первая буква имени гласная  $\wedge$  Четвертая буква имени гласная.

Этому условию удовлетворяет только имя АНТОН

Ответ: 3.

Второй способ реализует универсальный подход, но первый, в котором фактически используется таблица истинности импликации, более нагляден.

**Пример**

Каково наибольшее целое число  $X$ , при котором истинно высказывание  $(50 < X \cdot X) \rightarrow (50 > (X+1) \cdot (X+1))$ ?

Решение

Допустим, что  $X^2 > 50$ , тогда из математических соображений следует, что  $(X+1)^2 > 50$  и, следовательно, вся импликация ложна. Напротив, при  $X^2 \leq 50$  импликация всегда будет истинной, независимо от правой части (см. таблицу истинности импликации). Семь – Найдем наибольшее целое число  $X$ , такое что  $X^2 \leq 50$ . Очевидно  $X=7$

Ответ: 7

**Пример:**

Какое логическое выражение равносильно выражению  $\neg(A \vee \neg B)$ ?

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

Решение:

Воспользуемся тождествами:

$$\begin{aligned} \neg(a \vee b) &= (\neg a) \wedge (\neg b) \\ \neg(\neg b) &= b \end{aligned}$$

Получим высказывание:

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

Ответ: 4

**Пример:**

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

Дан фрагмент таблицы истинности выражения  $F$ :

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

Какое выражение соответствует  $F$ ?

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

**Решение:**

Чтобы не строить таблицу истинности для каждого выражения, можно просто пере-  
 проверить предложенные ответы:

- 1)  $\neg X \wedge \neg Y \wedge Z = 0$  при  $X = 0, Y = 0, Z = 0$ , что не соответствует первой строке таблицы;
- 2)  $\neg X \vee \neg Y \vee Z = 1$  при  $X = 0, Y = 0, Z = 1$ , что не соответствует второй строке таблицы.
- 3) выражение  $X \vee Y \vee \neg Z$  соответствует F, при всех предложенных комбинациях  $X, Y$  и  $Z$ ;
- 4)  $X \vee Y \vee Z = 1$  при  $X = 0, Y = 0, Z = 1$ , что не соответствует второй строке таблицы.

**Ответ:** 3

**Пример:**

Укажите значения переменных  $K, L, M, N$ , при которых логическое выражение  
 $(\neg K \vee M) \rightarrow (\neg L \vee M \vee N)$  **ложно**. Ответ запишите в виде строки из четырех символов:  
 значений переменных  $K, L, M$  и  $N$  (в указанном порядке) Так, например, строка 1101 со-  
 ответствует тому, что  $K=1, L=1, M=0, N=1$

**Решение:**

*Первый способ* решения подходит для конкретной задачи. Он основан на том, что им-  
 пликация  $x \rightarrow y$  ложна тогда и только тогда, когда  $x$  – истинно, а  $y$  – ложно.  $x = \neg K \vee M, y =$   
 $\neg L \vee M \vee N$ , отсюда по свойству дизъюнкции

$x$  истинно при  $(K, M) = (0, 0), (0, 1), (1, 1)$ ,

$y$  ложно лишь при  $(L, M, N) = (1, 0, 0) \Rightarrow M=0 \Rightarrow K=0$ .

Поэтому ответ 0100

*Второй способ* решения основан на тождественных преобразованиях. Он универ-  
 сальный, хотя и более громоздкий. Для данного случая воспользуемся тождествами:

$$a \rightarrow b = (\neg a) \vee b$$

$$\neg (a \vee b) = (\neg a) \wedge (\neg b)$$

$$\neg (\neg a) = a$$

Получим:

$$\begin{aligned} (\neg K \vee M) \rightarrow (\neg L \vee M \vee N) &= \neg (\neg K \vee M) \vee (\neg L \vee M \vee N) = \\ &= (K \wedge \neg M) \vee (\neg L \vee M \vee N) \end{aligned}$$

$(K \wedge \neg M) \vee (\neg L \vee M \vee N) = 0$  - для выполнения этого равенства необходимо, чтобы  
 $\neg L \vee M \vee N = 0$ , но это выполняется лишь при  $L = 1, M = 0, N = 0$

Также необходимо, чтобы  $K \wedge \neg M = 0$ , но при  $M = 0$  (т.е. при  $\neg M = 1$ ) это достигается  
 только при  $K = 0$ . Таким образом, получаем ответ:  $K = 0, L = 1, M = 0, N = 0$ , т.е. 0100

Если продолжать универсальный путь тождественных преобразований, то тогда обра-  
 тим внимание, что мы ищем значения переменных  $K, L, M, N$ , при которых логическое  
 выражение **ложно**. Проинвертируем это выражение (применим к нему операцию отрица-  
 ния) и будем искать значения переменных, при которых новое выражение **истинно**.

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

$$\text{Воспользуемся тождествами: } \neg (a \vee b) = (\neg a) \wedge (\neg b)$$

$$\neg ((K \wedge \neg M) \vee (\neg L \vee M \vee N)) = \neg (K \wedge \neg M) \wedge (\neg (\neg L \vee M \vee N)) =$$

$$\text{Воспользуемся тождествами: } \neg (a \wedge b) = (\neg a) \vee (\neg b)$$

$$\neg (\neg a) = a$$

$$= \neg (K \wedge \neg M) \wedge (\neg (\neg L \vee M \vee N)) = (\neg K \vee M) \wedge (L \wedge \neg M \wedge \neg N) =$$

$$= (\neg K \vee M) \wedge L \wedge \neg M \wedge \neg N$$

Это выражение принимает значение равное «1» только при условии:

$$\neg N = 1, \text{ т.е. } N = 0,$$

$$\neg M = 1, \text{ т.е. } M = 0,$$

$$L = 0, \text{ и}$$

$$\neg K \vee M = 1$$



Последнее выражение может равняться «1», при условии, что  $M = 0$ , только в одном случае, когда  $\neg K = 1$ , или  $K = 0$

В итоге получим:

$$K = 0, L = 1, M = 0, N = 0$$

Ответ: 0100

**Пример:**

$X, Y, Z$  - целые числа, для которых истинно высказывание

$$((Z < X) \vee (Z < Y)) \wedge \neg((Z+1) < X) \wedge \neg((Z+1) < Y)$$

Чему равно  $Z$ , если  $X=20, Y=10$ ?

**Решение:**

Преобразуем данное высказывание

$$((Z < X) \vee (Z < Y)) \wedge \neg((Z+1) < X) \wedge \neg((Z+1) < Y) = ((Z < X) \vee (Z < Y)) \wedge ((Z+1) \geq X) \wedge ((Z+1) \geq Y)$$

Подставим значения  $X$  и  $Y$ :

$$((Z < 20) \vee (Z < 10)) \wedge ((Z+1) \geq 20) \wedge ((Z+1) \geq 10) = ((Z < 20) \vee (Z < 10)) \wedge (Z \geq 19) \wedge (Z \geq 9)$$

Заметим, что  $(Z \geq 19) \wedge (Z \geq 9) = (Z \geq 19)$ , т.к., если истинно  $(Z \geq 19)$ , то истинно и  $(Z \geq 19) \wedge (Z \geq 9)$ , и наоборот, если  $(Z \geq 19)$  ложно, то ложно и  $(Z \geq 19) \wedge (Z \geq 9)$ .

Итак,

$$((Z < 20) \vee (Z < 10)) \wedge (Z \geq 19) \wedge (Z \geq 9) = ((Z < 20) \vee (Z < 10)) \wedge (Z \geq 19)$$

Воспользуемся распределительным законом

$$((Z < 20) \vee (Z < 10)) \wedge (Z \geq 19) = ((Z < 20) \wedge (Z \geq 19)) \vee ((Z < 10) \wedge (Z \geq 19))$$

Высказывание  $((Z < 10) \wedge (Z \geq 19))$  тождественно ложно, поэтому

$$((Z < 20) \wedge (Z \geq 19)) \vee ((Z < 10) \wedge (Z \geq 19)) = (Z < 20) \wedge (Z \geq 19)$$

Существует единственное целое число меньше 20 и, при этом большее или равное 19. Это 19

Ответ: 19.

**Пример:**

Три свидетеля дорожного происшествия сообщили сведения о скрывшемся нарушителе. Боб утверждает, что тот был на синем «Рено», Джон сказал, что нарушитель уехал на черной «Тойоте», а Сэм показал, что машина была точно не синяя 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 синего цвета»;

$B$  = «машина была «Рено»,

$C$  = «машина черного 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, получим:

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

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

$$\underline{B \wedge C \wedge \neg A} = 1.$$

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

**Ответ: РЕНО ЧЕРНЫЙ**

**Решение (способ 2).** Предположим, что машина была марки «Тойота». Тогда Джон ошибся в цвете, а Боб и Сэм ошиблись в марке, но верно определили цвет. Поскольку Боб и Сэм указали разные цвета, мы пришли к противоречию, следовательно, машина не «Тойота», и Джон ошибся в марке. Но тогда Джон верно указал цвет – черный. Значит, Боб ошибся в цвете, следовательно он верно указал марку – «Рено»

**Ответ: РЕНО ЧЕРНЫЙ.**

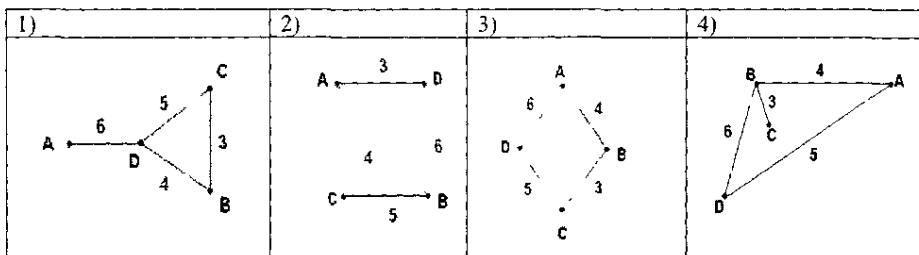
#### 1.4. Моделирование

При решении задач на эту тему необходимо уметь преобразовывать табличные модели (таблицы) в эквивалентные сетевые модели, представленные в виде схем. И, наоборот, по заданной схеме уметь строить эквивалентную табличную модель.

##### Пример:

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

	A	B	C	D
A		4		6
B	4		3	
C		3		5
D	6		5	



##### Решение:

Перепишем таблицу в следующем виде: пара соседних станций – стоимость перевозок между ними. Поскольку в данном случае стоимость перевозок между соседними станциями не зависит от направления, выписав пару XY, симметричную ей пару YX можно не писать.

AB – 4

AD – 6

BC – 3

DC – 5

У нас получился полный список элементов схемы. Только одна схема соответствует этому списку – схема под номером 3

**Ответ: 3.**

**Пример:**

Между четырьмя крупными аэропортами, обозначенными кодами DLU, IGT, OPK и QLO, ежедневно выполняются авиарейсы. Приведён фрагмент расписания перелетов между этими аэропортами:

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
QLO	IGT	06:20	08:35
IGT	DLU	10:25	12:35
DLU	IGT	11:45	13:30
OPK	QLO	12:15	14:25
QLO	DLU	12:45	16:35
IGT	QLO	13:15	15:40
DLU	QLO	13:40	17:25
DLU	OPK	15:30	17:15
QLO	OPK	17:35	19:30
OPK	DLU	19:40	21:55

Путешественник находится в аэропорту DLU в полночь (0:00). Определите самое раннее время, когда он может оказаться в аэропорту QLO.

- 1) 15:40                      2) 16:35                      3) 17:15                      4) 17:25

**Решение:**

Выпишем возможные цепочки перелетов, исключая циклические:

DLU-> IGT, IGT->QLO (11:45->13:30, 13:15->15:40)

DLU->QLO (13:40->17:25);

DLU->OPK, OPK->QLO (15:30->17:15, 12:15->14:25).

Вторая цепочка обеспечивает самый быстрый перелет, поскольку в остальных случаях в аэропорте пересадки придется провести почти сутки. Поэтому ответ - 17:25.

Следует заметить, что если бы первая цепочка выглядела бы, например, так:

DLU -> IGT, IGT->QLO (11:45->13:30, 14:30->16:40), то путь с пересадкой оказался бы быстрее прямого перелета.

## 2. Информационные и коммуникационные технологии

### 2.1. Файловые системы

Во всех операционных системах имеющаяся на компьютере информация хранится в виде файлов. Каждый файл имеет имя и располагается на определенном устройстве хранения информации. В виде файлов хранятся программы (такие файлы называются выполняемыми) и документы. Иногда в состав одного приложения или документа входят несколько файлов. Для удобства хранения и поиска файлов они объединены в папки. Папки могут быть вложены друг в друга, образуя многоуровневую древовидную структуру.

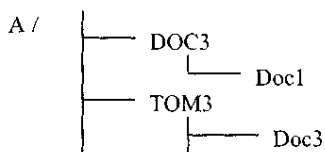
Синонимами термина “папка”, принятого в Windows, являются слова “каталог” и “директория”. Подобно файлам папки имеют свои имена.

Полное имя файла состоит из пути к файлу и имени файла. Путь к файлу представляет собой перечень имен каталогов (папок), которые нужно последовательно посетить, чтобы спуститься к файлу с самого высокого уровня дерева файлов. Во многих операционных системах (в том числе семейства Windows) полное имя файла начинается с имени устройства внешней памяти, на котором он расположен. Имя устройства образуется двумя символами – латинской буквой и следующим за ней двоеточием. После имени устройства следуют имена папок, разделенные символом «\» или «/».

Рассмотрим решение типовых задач по этой теме.

#### Пример:

Дано дерево каталогов.



Определите полное имя файла Doc3.

- 1) A:/DOC3
- 2) A:/DOC3/Doc3
- 3) A./ DOC3/Doc1
- 4) A./TOM3/Doc3

#### Решение:

Путь к файлу Doc3 состоит из следующих элементов.

Имя устройства внешней памяти – A

Имя каталога – TOM3

Таким образом, полное имя файла Doc3 – A:/TOM3/Doc3, что соответствует пункту 4 в списке ответов

#### Пример:

В некотором каталоге хранился файл **txt.doc**. После того, как в этом каталоге создан подкаталог и переместили в созданный подкаталог файл **txt.doc**, полное имя файла стало **P:\doc\txt\letter\txt.doc**. Каково было полное имя этого файла до перемещения?

#### Решение:

По условию задачи файл должен находиться во вновь созданном каталоге. По полному имени файла видно, что он находится в каталоге **letter**, следовательно, **letter** и есть вновь созданный каталог. Каталог **letter** находится в каталоге с полным именем **P:\doc\txt**.

По условию задачи файл изначально хранился в том каталоге, где был создан подкаталог **letter**, следовательно, полное имя файла было **P:\doc\txt\txt.doc**

**Пример:**

Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы

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

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

Определите, какое из указанных имен файлов удовлетворяет маске:

**?hel\*lo.c?\***.

- 1) hello c                      2) hello.cpp                      3) hhelolo.cpp                      4) hhelolo.c

**Решение:**

Поскольку в начале маски стоит знак «**?**» перед буквой **h**, то в имени буква **h** должна стоять на втором месте. Поскольку в маске после буквы **c** тоже стоит знак «**?**», то после этой буквы в имени файла должен находиться непустой символ. Этим условиям из предложенных имен файлов одновременно удовлетворяет только третье - **hhelolo.cpp**. Нетрудно убедиться, что оно также соответствует остальным символам маски

**Ответ: 3**

## 2.2. Обработка графической информации

Для успешного решения задач этой темы необходимо понимать взаимосвязь информационного объема растрового изображения, его пространственного и цветового разрешения, а также принципы кодирования цвета элементов растра.

**Пример:**

Для хранения растрового изображения размером 64х64 пикселя отвели 1,5 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

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

**Решение:**

Вычислим, сколько бит отводится для хранения одного пикселя, учитывая, что в одном килобайте  $1024 = 2^{10}$  байта, а в одном байте  $8 = 2^3$  бита. Для этого информационный объем изображения, выраженный в битах, разделим на количество пикселей.

$$1,5 \cdot 1024 \cdot 8 / (64 \cdot 64) = 1,5 \cdot 2^{10} \cdot 2^3 / (2^6 \cdot 2^6) = 1,5 \cdot 2^{10+3-6-6} = 1,5 \cdot 2^1 = 3.$$

Итак, на один пиксель приходится три бита. Три бита позволяют закодировать максимум  $2^3 = 8$  различных значений. Поэтому максимально возможное число цветов в палитре изображения равно 8, что соответствует ответу под номером 1

**Ответ: 1.**

**Пример:**

Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 256х256 пикселей, если известно, что в изображении используется палитра из 216 цветов. Саму палитру хранить не нужно.

- 1) 128                      2) 512                      3) 1024                      4) 2048

**Решение:**

Найдем минимальный объем памяти, необходимый для хранения одного пикселя. В изображении используется палитра из  $2^{16}$  цветов, следовательно, одному пикселю может быть сопоставлен любой из  $2^{16}$  возможных номеров цвета в палитре. Поэтому минимальный объем памяти для одного пикселя будет равен  $\log_2(2^{16}) = 16$  битам. Минимальный объем памяти, достаточный для хранения всего изображения будет равен

$16 \cdot 256 \cdot 256 = 2^4 \cdot 2^8 \cdot 2^8 = 2^{20}$  бит =  $2^{17}$  байт = 2 килобайт = 128 килобайт, что соответствует пункту под номером 1.

**Ответ: 1.**

**Пример:**

Для кодирования цвета фона страницы Интернет используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом `<body bgcolor="#FFFFFF">?`

- 1) белый                      2) зеленый                      3) красный                      4) синий

**Решение:**

В 24-битной системе кодирования цветов RGB, используемой для мониторов каждая из трех цветовых составляющих (красная, зеленая, синяя) может принимать значение от 0 до 255 и кодируется одним байтом (две шестнадцатеричные цифры). 0 соответствует полному отсутствию данной цветовой составляющей в пикселе, 255 ( $FF_{16}$ ) – максимальной яркости данного цвета. Максимальная яркость всех трех цветовых сигналов  $FFFFFF_{16}$  в RGB обеспечивает белый цвет пикселя. Минимальная (все цвета выключены) –  $000000_{16}$  – черный.

**Ответ: 1**

### 2.3. Обработка информации в электронных таблицах

#### *Работа с формулами, абсолютные и относительные ссылки*

Электронная таблица состоит из ячеек, каждая из которых имеет адрес аналогично клеточкам шахматной доски или игрового поля для "морского боя". Столбцы таблицы нумеруются буквами и буквосочетаниями латинского алфавита, а строки – числами. Таким образом, ячейка на пересечении 15-й строки и столбца D будет иметь адрес D15.

В ячейке может храниться не только значение, введенное пользователем, но и формула, по которой автоматически рассчитывается значение ячейки. Для того, чтобы электронная таблица могла производить вычисления по формулам, они должны быть записаны по определенным правилам. Отличительным признаком формулы является знак = в ее начале. В формулах могут использоваться адреса других ячеек и различные функции, в том числе математические  $\sin$ ,  $\cos$ ,  $\exp$  (показательная функция с основанием, равным числу  $e$ ),  $\text{abs}$  (модуль числа) и так далее.

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

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

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

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

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

### Пример:

В ячейке A1 электронной таблицы записана формула  $=D1-\$D2$ . Какой вид приобретет формула после того, как ячейку A1 скопируют в ячейку B1?

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

### Решение:

Адрес B1 получен из адреса A1 сдвигом на 1 вправо. Также изменятся все относительные адреса столбцов в адресах формулы, а именно, D1 преобразуется в E1. Адреса строк не изменятся, так как формула копируется в пределах одной строки (первой). Адрес  $\$D2$  не изменится, так как здесь адрес столбца абсолютный, и формула приобретет вид  $=E1-\$D2$ .

**Ответ:** 2

### Пример:

Дан фрагмент электронной таблицы:

	A	B	C	D
1	5	2	4	
2	10	1	6	

В ячейку D2 введена формула  $=A2*B1+C1$

В результате в ячейке D2 появится значение:

- 1) 6      2) 14      3) 16      4) 24

### Решение:

$$D2 = 10 \cdot 2 + 4 = 24$$

**Ответ:** 4

### Пример:

Дан фрагмент электронной таблицы:

	A	B	C	D
1	5	2	4	
2	10	1	6	

В ячейку D2 введена формула  $=A2*B1+C1$

В результате в ячейке D2 появится значение:

- 1) 6      2) 14      3) 16      4) 24

**Решение:**

Подставим значения ячеек в заданную формулу:  $A2 \cdot B1 + C1 = 10 \cdot 2 + 4 = 24$

**Ответ:** 4.

**Пример:**

В электронной таблице значение формулы =СУММ(B1:B2) равно 5. Чему равно значение ячейки B3, если значение формулы =СРЗНАЧ(B1:B3) равно 3?

**Решение:**

Диапазону B1:B3 принадлежат три ячейки B1, B2 и B3. Формула СРЗНАЧ вычисляет среднее арифметическое ячеек диапазона, поэтому  $(B1 + B2 + B3) / 3 = 3$ . Следовательно  $B1 + B2 + B3 = 9$ . Учитывая, что по условию  $B1 + B2 = 5$ , получаем, что  $B3 = 4$ .

**Диаграммы и графики**

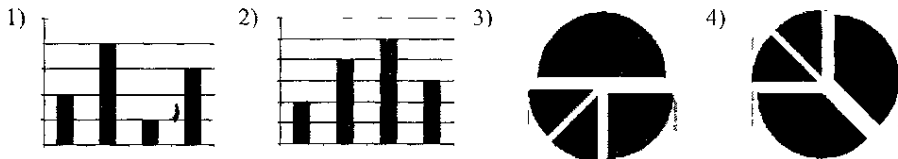
Для решения задач этой темы важно уметь правильно интерпретировать столбчатые и круговые диаграммы, учитывая, что простые столбчатые диаграммы обычно отражают абсолютные значения, а круговые – процентный состав.

**Пример:**

Дан фрагмент электронной таблицы:

	A	B
1	=A4-B1	1
2	=A1+2	2
3	=A2+B1	
4	=B1+B2	

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

**Решение:**

В диапазон A1:A4 входят ячейки A1, A2, A3, A4. Прежде всего, следует вычислить значения ячеек указанного диапазона. Мы можем вычислить значение формулы только в том случае, если известны значения всех фигурирующих в ней ячеек. Поэтому, порядок заполнения таблицы будет таков.

	A	B
1	=A4-B1	1
2	=A1+2	2
3	=A2+B1	
4	3	

	A	B
1	2	1
2	=A1+2	2
3	=A2+B1	
4	3	



	A	B
1	2	1
2	4	2
3	$=A2+B1$	
4	3	

	A	B
1	2	1
2	4	2
3	5	
4	3	

Итак, значения ячеек диапазона A1:A4 равны 2; 4; 5; 3 соответственно. Такому набору значений соответствует диаграмма под номером 2.

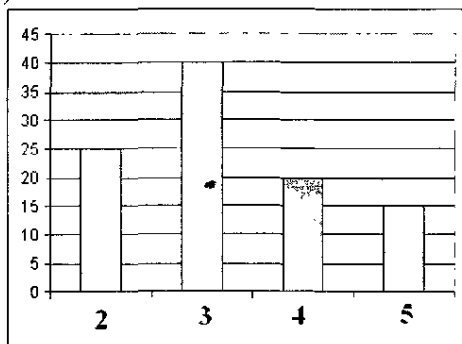
**Ответ: 2.**

### Пример:

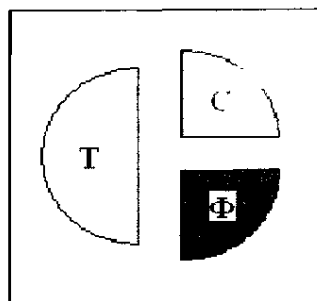
В цехе трудятся рабочие трех специальностей – токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд не меньший второго и не больший пятого. На диаграмме I отражено количество рабочих с различными разрядами, а на диаграмме II – распределение рабочих по специальностям.

Каждый рабочий имеет только одну специальность и один разряд

I)



II)



Имеются четыре утверждения:

А) Все рабочие третьего разряда могут быть фрезеровщиками

Б) Все слесари могут быть пятого разряда

В) Все токари могут быть четвертого разряда

Г) Все рабочие третьего разряда могут быть токарями

Какое из этих утверждений следует из анализа обеих диаграмм?

1) А

2) Б

3) В

4) Г

### Решение:

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

Из правой диаграммы следует, что токарей половина от общего количества рабочих (т.е.  $0.5 \cdot 100 = 50$  чел.), слесарей и фрезеровщиков – по 25 чел

Проверим утверждение А. Оно ложно, так как рабочих третьего разряда 40 чел., а фрезеровщиков всего 25, следовательно **все** рабочие третьего разряда не могут быть фрезеровщиками

Проверим утверждение Б. Слесарей 25 чел., рабочих пятого разряда всего 15 чел. Поэтому все слесари не могут иметь пятый разряд. Утверждение Б ложно.

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

Утверждение Г – истинно, так как рабочих третьего разряда насчитывается 40 человек, следовательно все они могут входить в число токарей, которых 50 чел

**Ответ: 4**

#### 2.4. Базы данных

В задачах на эту тему обычно требуется вручную выполнить заданный поисковый запрос к предложенному фрагменту базы данных или самостоятельно сформулировать такой запрос исходя из поставленной цели.

Разберем примеры таких задач

##### Пример:

Сколько записей в нижеследующем фрагменте турнирной таблицы удовлетворяют условию «Место <=4 И (В>4 ИЛИ МЗ>12)»?

Место	Команда	В	Н	П	О	МЗ	МП
1	Боец	5	3	1	18	9	5
2	Авангард	6	0	3	18	13	7
3	Опушка	4	1	4	16	13	7
4	Звезда	3	6	0	15	5	2
5	Химик	3	3	3	12	14	17
6	Пират	3	2	4	11	13	7

1) 5

2) 2

3) 3

4) 4

##### Решение:

Решая данную задачу, необходимо последовательно применять условие к каждой строке таблицы. Условию удовлетворяют команды: Боец, Авангард, Опушка. У Звезды недостаточно выигрышей и забитых мячей, Химик и Пират заняли слишком низкие места.

**Ответ: 3.**

##### Пример:

Для каждого файла в таблицу записывался исходный размер файла (поле РАЗМЕР), а также размеры архивов, полученных после применения к файлу различных архиваторов: программы WinZIP (поле ZIP), программы WinRAR (поле RAR) и программы StuffIt (поле SIT). Вот начало этой таблицы (все размеры в таблице - в килобайтах)

Имя файла	РАЗМЕР	ZIP	RAR	SIT
Аквариум.mw2	296	124	88	92
Муар.mw2	932	24	20	28

Нужно отобрать файлы, исходный размер которых больше 1 мегабайта и размер которых при использовании WinRAR уменьшился более чем в 4 раза. Для этого достаточно найти в таблице записи, удовлетворяющие условию:

- 1) (РАЗМЕР > 1000) ИЛИ (РАЗМЕР / RAR > 4)
- 2) (РАЗМЕР > 1024) И (RAR < 256)
- 3) (РАЗМЕР > 1024) И (РАЗМЕР / RAR > 4)
- 4) (РАЗМЕР > 1024) ИЛИ (РАЗМЕР / RAR > 4)

**Решение.** В задачах этого типа конкретные значения полей базы данных не важны, таблица приводится только для визуализации описанной структуры базы данных. Задача состоит в том, чтобы сформировать правильный запрос по описанному условию. Ключевым здесь является условие конъюнкции (логического умножения) требований, а также соотношение 1 Мбайт = 1024 Кбайт

**Ответ:** 3.

## 2.5. Телекоммуникационные технологии

### *Адресация в сети Интернет*

Web-страницы в Интернет размещаются на Web-серверах в виде текстовых файлов. Входящие в состав страниц изображения, звуки и другие компоненты тоже размещаются на сервере в виде отдельных файлов. Файлы страниц и их компонентов имеют свои адреса (идентификаторы), называемые унифицированными указателями ресурсов (URL — Uniform Resource Locator)

**Пример URL-адреса:** <http://www.mai.ru/chair806/index.htm>

URL-адрес начинается с обозначения протокола Интернета, с помощью которого осуществляется доступ к ресурсу. В данном случае указан HTTP — протокол передачи гипертекста (HyperText Transfer Protocol). После названия протокола следуют двоеточие и две наклонных черты. Далее указывается имя компьютера ([www.mai.ru](http://www.mai.ru)). После имени компьютера следует полное имя файла с ресурсом на сервере, символы "/" разделяют названия каталогов, которые нужно посетить, чтобы добраться до файла. В данном случае страница хранится на сервере в файле `index.htm` каталога `chair806`.

### **Пример:**

Идентификатор некоторого ресурса сети Интернет имеет следующий вид:  
**<http://www.mail.ru/ftp.html>**

Какая часть этого идентификатора указывает на протокол, используемый для передачи ресурса?

- |                     |                     |                      |                      |
|---------------------|---------------------|----------------------|----------------------|
| 1) <code>www</code> | 2) <code>ftp</code> | 3) <code>http</code> | 4) <code>html</code> |
|---------------------|---------------------|----------------------|----------------------|

### **Решение:**

Название протокола записывается в самом начале идентификатора ресурса. В данном случае это **`http`**.

**Ответ:** 3

### **Пример:**

Доступ к файлу `net.edu`, находящемуся на сервере `ru.com` осуществляется по протоколу `ftp`

В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите по следовательность этих букв, кодирующую адрес указанного файла.

А	ftp
Б	ru
В	://
Г	.edu
Д	com
Е	net
Ж	/

**Решение:**

Имя ресурса начинается с названия протокола, в данном случае – это ftp (буква А). Имя протокола должно отделяться от имени сервера двоеточием и двумя наклонными чертами (В) Имя сервера – ru com кодируется буквами Б и Д. После имени сервера следует наклонная черта (Ж), отделяющая его от имени файла (Е,Г). Итак, имя ресурса ftp://ru com/net edu

**Ответ:** АВБДЖЕГ

**Пример:**

Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г Восстановите IP-адрес

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

			
<b>А</b>	<b>Б</b>	<b>В</b>	<b>Г</b>

**Решение**

IP-адрес не может содержать четырех- или пятизначное число, поэтому фрагмент Г должен быть на первом месте. Он также не может содержать трехзначное число 643 (643>255), поэтому фрагмент А – на последнем месте Учитывая, что число 1333 тоже не может входить в его состав, получаем искомый IP-адрес 203.133.133 64 (ГБВА)

**Ответ:** ГБВА

**Поиск информации в сети Интернет**

Для более точной формулировки запросов к поисковому используются логические связки И и ИЛИ. Связка И между двумя словами означает, что требуется найти страницы, содержащие одновременно и первое, и второе слово. Связка ИЛИ – что ищутся страницы, включающие хотя бы одно из указанных слов

**Пример**

Каким условием нужно воспользоваться для поиска в сети Интернет информации о цветах, растущих на островах Тайвань или Хонсю (для обозначения логической операции «ИЛИ» используется символ |, а для операции «И» - символ &)?

- 1) цветы&(Тайвань|Хонсю)
- 2) цветы&Тайвань&Хонсю

- 3) цветы|Тайвань|Хонсю  
 4) цветы&(остров|Тайвань|Хонсю)

### Решение:

По смыслу задачи, в любой найденной странице должно содержаться слово «цветы» и название одного из двух указанных островов. Этому условию удовлетворяют запросы 1 и 4.

Запрос 2 требует одновременного присутствия на странице слов «Тайвань» и «Хонсю», что является избыточным требованием и неоправданно сужает поле для поиска. Запрос 3, напротив, его неоправданно расширяет. По запросу 3 будет найдено, например, много страниц о цветах, на которых не будет ни слова «Тайвань», ни слова «Хонсю».

Итак, выбираем между запросами 1 и 4. В запросе 4 наравне с «Тайвань» и «Хонсю» присутствует слово «остров», что также излишне расширяет круг поиска страницами о цветах и островах. Поэтому, правильный ответ – 1.

**Ответ:** 1.

### Пример:

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

Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» - &

А	законы & физика
Б	законы   (физика & биология)
В	законы & физика & биология & химия
Г	законы   физика   биология

### Решение:

Приведем два способа решения, один из которых основан на рассуждении, а второй предполагает использование графического представления операций над множествами. Рассуждая логически, мы видим, что больше всего будет найдено страниц по запросу Г, так как при его исполнении будут найдены и страницы со словом «законы» (в том числе, например, и юридические), и страницы, со словом «физика», и страницы со словом «биология». Меньше всего будет найдено страниц по запросу В, так как в нем требуется присутствие всех четырех слов на искомой странице.

Осталось сравнить запросы А и Б. По запросу Б будут найдены все страницы, соответствующие запросу А, (так как в последних обязательно присутствует слово «законы»), а также страницы, содержащие одновременно слова « физика » и « биология ». Следовательно, по запросу Б будет найдено больше страниц, чем по запросу А

Итак, упорядочив запросы по возрастанию страниц, получаем ответ ВАБГ.

Для решения вторым способом рассмотрим множества страниц, содержащие каждое из искомых слов. Запросу X&Y будет соответствовать пересечение множеств X и Y, а запросу X | Y – их объединение. 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:



Для Б:  
законы | (физика & биология)



Для В:  
законы & физика & биология & химия



Для Г:  
законы | физика | биология



Упорядочив четыре полученные диаграммы по степени закрашенности, получаем ответ: ВАБГ.

Ответ : ВАБГ

## 2.6. Технология программирования

В задании С1 демо-версии 2007г. проверяется умение прочесть программу, предназначенную для решения определенной задачи, и исправить допущенные программистом логические ошибки.

Прежде всего, следует полностью уяснить задачу, которую должна решать программа. Без этого поиск ошибок в решении не имеет смысла. После того, как выяснена цель программы, можно приступить к анализу ее текста. Не рекомендуется 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.

Итак, выбираем наиболее знакомый язык программирования и пытаемся 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 следует пытаться искать синтаксические ошибки, то есть 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.п.) Речь идет только об ошибках в логике алгоритма. Если не удастся сходу их найти, то можно попробовать выполнить программу 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.

Некоторые учащиеся поступают следующим образом. пишут свою программу решения задачи, а затем сравнивают с приведенным в условии текстом. Такой подход тоже возможен.

Чтобы убедиться, что ошибка действительно обнаружена, 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 работала неправильно, и при этом не утратила способности правильно работать для тех данных, с которыми она верно работала и до внесения исправлений.

При подборе тестовых исходных данных следует руководствоваться следующими правилами.

- 1) Обязательно следует рассматривать нулевые, граничные и прочие "критические" значения исходных данных (если такие значения могут иметь место)
- 2) Тестовые данные должны включать такие комбинации исходных данных, чтобы обеспечивалось выполнение всех исполняемых операторов программы, в том числе ветвлений и циклов.

### Пример:

В программе вводятся с клавиатуры три различных целых числа в переменные **a**, **b**, **c**. Далее в программе требуется преобразовать исходные данные так, чтобы в переменной **a** оказалось наименьшее число, в переменной **b** оказалось среднее число, в переменной **c** - наибольшее число.

Программа на Паскале	Программа на Бейсике
<pre>var   a, b, c, x: integer; begin   write('Введите a,b,c'); read(a,b,c);   if a&gt;b then begin x:=a; a:=b; b:=x end;   if b&gt;c then begin x:=b; b:=c; c:=x end end</pre>	<pre>PRINT " Введите a,b,c" INPUT a,b,c IF a&gt;b THEN x=a a=b b=x IF b&gt;c THEN x=b. b=c: c=x</pre>

Программист допустил ошибку при написании программы.

1) Привести пример таких исходных значений **a**, **b**, **c**, после обработки которых исходной программой соотношение  $a < b < c$  не будет выполнено.

2) Исправить ошибку в программе, так чтобы она полностью соответствовала постановке задачи.

**Решение:**

1)  $a=2$ ;  $b=3$ ;  $c=1$  - после обработки такого набора значений  $a, b, c$  исходной программой соотношение  $a < b < c$  не будет выполнено, следовательно, в конце программы будет  $a=2$ ;  $b=1$ ,  $c=3$ .

(Другой пример.  $a=3$ ;  $b=2$ ;  $c=1$  - такие исходные данные дают аналогичный результат: невыполнение соотношения  $a < b < c$  на выходе программы, т.к. на выходе получим  $a=2$ ;  $b=1$ ,  $c=3$ .)

2) Для доработки программы до правильной нужно добавить еще одну строчку, так чтобы обработка состояла из операторов.

```
if a>b then begin x:=a; a:=b; b:=x end;  
if b>c then begin x:=b; b:=c; c:=x end;  
if a>b then begin x:=a; a:=b; b:=x end;
```

- это самый простой вариант доработки.

(На первый взгляд может показаться, что последняя строчка является бессмысленным повторением первой, однако это не так - ведь исходная программа обеспечивала на выходе только то, что в переменной  $c$  находилось максимальное число, а значения  $a$  и  $b$ , как это видно из приведенных примеров, могли не удовлетворять условию  $a < b$ ; добавление же последней строчки обеспечивает то, что на выходе программы условие  $a < b$  будет выполнено )

В задании С4 демо-версии 2007г. требуется написать программу для решения определенной задачи. Программа должна быть по возможности эффективной, то есть не выполнять лишний раз те действия, которые можно выполнить однократно, и не хранить дополнительные данные, без которых можно обойтись.

**Пример:**

Во входном файле `meteo.dat` 365 строк, которые содержат информацию о среднесуточной температуре всех дней 2003 г. 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 записано значение температуры — число со знаком плюс или минус, с точностью до 1 цифры после десятичной точки. Данная информация отсортирована по значению температуры, т.е. хронологический порядок нарушен. Требуется написать программу на языке Паскаль или Бейсик, которая рассчитает и выведет на экран следующую выходную информацию.

1) среднегодовую температуру (в отдельной строчке вывода);

2) информацию о месяцах, для которых абсолютная величина разности среднемесячной и среднегодовой температуры не превосходит 5 градусов. Значения для каждого из месяцев, удовлетворяющих этому условию, следует выводить в отдельной строке в виде: номер месяца, значение среднемесячной температуры.

**Решение:**

В качестве решения приводим текст программы на языке Паскаль.



```

Const d:array[1..12] of integer =
  (31,28,31,30,31,30,31,31,30,31,30,31);
var tm:array[1..12] of real;
    i,x:integer;
    data:string[5];
    ty,t:real;
begin
  assign(input,'meteo.dat');
  reset(input);
  for i:=1 to 12 do
    tm[i]:=0;
  ty:=0;
  for i:=1 to 365 do
    begin
      readln(data,t);
      x:=(ord(data[4])-ord('0'))*10
        +ord(data[5])-ord('0');
      tm[x]:=tm[x]+t;
      ty:=ty+t;
    end;
  for i:=1 to 12 do
    tm[i]:=tm[i]/d[i];
  ty:=ty/365;  writeln(' ty= ',ty:4:2);
  for i:=1 to 12 do
    if abs(tm[i]-ty) < 5 then writeln(i,tm[i]:5:2);
  end.

```

#### **Пояснения:**

Программа читает текст из файла один раз, сразу подсчитывая в массиве, хранящем 12 вещественных чисел, сумму температур в каждом из месяцев, одновременно суммируя все температуры в году. Затем производится дальнейшая обработка для получения выходных результатов и вывода их на экран (смысл этой обработки понятен из конечных строк программы).

За эту программу можно получить максимальную оценку (4 балла), если программа написана правильно и достаточно эффективно – в первую очередь это означает, что цикл от 1 до 365 проходится не более одного раза. Если у вас программа написана правильно, но недостаточно эффективно, то вы за эту задачу получите сколько-то баллов, но меньше четырех.

**Разберем еще несколько простых задач с приведением решения на языках Паскаль и Бейсик.**

#### **Пример:**

Требуется найти сумму  $N$  первых слагаемых ряда

$$a_1 + a_2 + a_3 + \dots$$

где член ряда задается выражением  $a_k = 1/k^2$ .

**Решение:**

Программа на Паскале

```

VAR s: real;
N, i: integer;
BEGIN
  write( 'Ввод N' ); read (N);
  s:=0;
  for i:=1 to N do
    s:= s + 1 / (i * i);
  write('s=',s);
END.

```

Программа на Бейсике

```

INPUT "N="; N
s = 0!
FOR i = 1 TO N
  s = s + 1 / (i * i)
NEXT i
PRINT "s=", s

(восклицательный знак означает
спецификацию числа s как веще-
ственного)

```

**Пример:**

Требуется найти сумму слагаемых ряда

$$a_1 + a_2 + a_3 + \dots$$

где член ряда задается выражением  $a_k = 1/k^2$ , а суммирование нужно выполнить с точностью  $\varepsilon$  (это означает, что суммирование нужно прекратить, если очередной член ряда меньше  $\varepsilon$ , где  $\varepsilon$  – малое число)

**Решение:**

Программа на Паскале

```

VAR s, eps: real;
      i: integer;
BEGIN
  write( 'ввод eps' ); read
  (eps);
  s:=0; i:=1;
  while 1 / (i * i) = >eps do
  begin
    s:= s + 1 / (i * i);
    i := i + 1
  end;
  write('s=',s);
END.

```

Программа на Бейсике

```

INPUT "eps="; eps
s = 0! : i = 1
WHILE 1 / (i * i) = >eps
  s = s + 1 / (i * i)
  i = i + 1
WEND
PRINT "s=", s

```

## ТРЕНИРОВОЧНЫЕ ЗАДАНИЯ

В этой главе приводятся тренировочные задания из трех источников: открытого сегмента федерального банка тестовых заданий, демонстрационных вариантов ЕГЭ прошлых лет и учебно-тренировочные материалы из книги “ЕГЭ Информатика” (Крылов С.С., Лещинер В.Р., Супрун П.Г., Якушкин П.А.; Под ред. Лещинера В.Р. – М.: Интеллект-Центр, 2005, 2006).

В соответствии с Кодификатором элементов содержания по информатике, задания сгруппированы в два раздела: “информационные процессы и системы” и “Информационные и коммуникационные технологии”. В каждом разделе задания имеют сквозную нумерацию.

### 1. Информационные процессы и системы

#### 1.1. Информация и ее кодирование

*Единицы измерения и методы измерения количества информации. Скорость передачи информации*

##### 1.1

Считая, что каждый символ кодируется одним байтом, оцените информационный объем следующего предложения:

*Мой дядя самых честных правил, Когда не в шутку занемог, Он уважать себя заставил И лучше выдумать не мог.*

- 1) 108 бит                      2) 864 бит                      3) 108 килобайт                      4) 864 килобайт

##### 1.2

Считая, что каждый символ кодируется двумя байтами, оцените информационный объем следующего предложения в кодировке Unicode:

*Один пуд – около 16,4 килограмм.*

- 1) 32 килобайта                      2) 512 бит                      3) 64 бита                      4) 32 байта

##### 1.3

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 60 байт. Какова длина сообщения в символах?

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

##### 1.4

Считая, что каждый символ кодируется одним байтом, оцените информационный объем следующего предложения из пушкинского четверостишия:

*Певец-Давид был ростом мал, Но повалил же Голиафа!*

- 1) 400 бит                      2) 50 бит                      3) 400 байт                      4) 5 байт

### 1.5

Каждый символ закодирован одним байтом. Оцените информационный объем следующего предложения в этой кодировке:

***В одном килограмме 1000 грамм.***

- 1) 16 килобайтов      2) 256 бит      3) 32 бита      4) 16 байтов

### 1.6

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 различных сигналов?

- 1) 8      2) 9      3) 100      4) 200

### 1.7

Одна ячейка памяти троичной ЭВМ (компьютера, основанного на троичной системе счисления) может принимать одно из трех возможных состояний. Для хранения некоторой величины отведи 4 ячейки памяти. Сколько различных значений может принимать эта величина?

- 1) 12      2) 16      3) 64      4) 81

### 1.8

Шахматная доска состоит из 64 полей. 8 столбцов на 8 строк. Какое минимальное количество бит потребуется для кодирования координат одного шахматного поля?

- 1) 4      2) 5      3) 6      4) 7

### 1.9

Сколько существует различных последовательностей из символов «а» и «б», длиной ровно в 10 символов?

- 1) 20      2) 100      3) 1024      4) 2048

### 1.10

Сколько существует различных последовательностей из символов «плюс» и «минус», длиной ровно в пять символов?

- 1) 64      2) 50      3) 32      4) 20

### 1.11

В зрительном зале две прямоугольные области зрительских кресел: одна 10 на 12, а другая 17 на 8. Какое минимальное количество бит потребуется для кодирования каждого места в автоматизированной системе?

### 1.12

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 трех и не более пяти сигналов (точек и тире)?

- 1) 12                      2) 56                      3) 84                      4) 256

**1.13**

Сколько мегабайт информации содержит сообщение объемом  $2^{23}$  бит?

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

**1.14**

Получено сообщение, информационный объем которого равен 32 битам. Чему равен этот объем в байтах?

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

**1.15**

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

A	B	C	D	E
000	01	100	10	011

Определить, какой набор букв закодирован двоичной строкой 0110100011000

- 1) EBCEA                      2) BDDEA                      3) BDCEA                      4) EBAEA

**1.16**

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 перевести результат в шестнадцатеричную систему счисления, то получится:

- 1) D2                      2) 132                      3) 3102                      4) DBAC

**1.17**

Каждое показание датчика, фиксируемое в памяти компьютера, занимает 10 бит. Записано 100 показаний этого датчика. Каков информационный объем снятых значений в байтах?

- 1) 10                      2) 100                      3) 125                      4) 1000

**1.18**

Метеорологическая станция ведет наблюдение за направлением ветра. Результатом одного измерения является одно из 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

### 1.19

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. Определите информационный объем результатов наблюдений.

- 1) 60 байт                      2) 70 бит                      3) 80 байт                      4) 480 байт

### 1.20

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

### 1.21

В велокроссе участвуют 119 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 70 велосипедистов?

- 1) 70 бит                      2) 70 байт                      3) 490 бит                      4) 119 байт

### 1.22

Обычный дорожный светофор без дополнительных секций подает шесть видов сигналов (непрерывные красный, желтый и зеленый, мигающие желтый и зеленый, красный и желтый одновременно). Электронное устройство управления светофором последовательно воспроизводит записанные сигналы. Подряд записано 100 сигналов светофора. В байтах данный информационный объем составляет

- 1) 37                              2) 38                              3) 50                              4) 100

### 1.23

Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 800x600 пикселей, при условии, что в палитре 16 миллионов цветов?

### 1.24

Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 640x480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?

### 1.25

Известно, что длительность непрерывного подключения к сети Интернет с помощью модема для некоторых АТС не превышает 10 минут. Определите максимальный размер файла (в килобайтах), который может быть передан за время такого подключения, если модем передает информацию в среднем со скоростью 32 килобит/с

**1.26**

Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать 100 страниц текста в 30 строк по 60 символов каждая, при условии что каждый символ кодируется одним байтом?

**1.27**

Сколько секунд потребуется модему, передающему сообщения со скоростью 14400 бит/с, чтобы передать сообщение длиной 225 Кбайт?

**Представление числовой информации. Системы счисления****1.28**

Сколько единиц в двоичной записи числа 197?

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

**1.29**

Количество значащих нулей в двоичной записи числа 129 равно:

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

**1.30**

Как представляется число  $25_{10}$  в двоичной системе счисления?

- 1)  $1001_2$                       2)  $11001_2$                       3)  $10011_2$                       4)  $11010_2$

**1.31**

Как представлено число  $83_{10}$  в двоичной системе счисления?

- 1)  $1001011_2$                       2)  $1100101_2$                       3)  $1010011_2$                       4)  $101001_2$

**1.32**

Дано  $a=D7_{16}$ ,  $b=331_8$ . Какое из чисел  $c$ , записанных в двоичной системе, отвечает условию  $a < c < b$ ?

- 1)  $11011001$                       2)  $11011100$                       3)  $11010111$                       4)  $11011000$

**1.33**

Чему равна разность чисел  $101_{16}$  и  $1100101_2$ ?

- 1)  $44_8$                       2)  $234_8$                       3)  $36_{16}$                       4)  $60_{16}$

**1.34**

Вычислите значение суммы в десятичной системе счисления:

$$10_2 + 10_8 + 10_{16} = ?_{10}$$

- 1)  $30_{10}$                       2)  $26_{10}$                       3)  $36_{10}$                       4)  $20_{10}$

**1.35**

Вычислите сумму чисел  $x$  и  $y$ , если

$$x=1110101_2;$$

$$y=1011011_2.$$

Результат представьте в виде восьмеричного числа.

- 1)  $2110_8$                       2)  $298_8$                       3)  $320_8$                       4)  $318_8$

**1.36**

Вычислите сумму двоичных чисел  $x$  и  $y$ , если

$$x = 1010101_2$$

$$y = 1010011_2$$

- 1)  $10100010_2$       2)  $10101000_2$       3)  $10100100_2$       4)  $10111000_2$

**1.37**

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)  $10111010_2$

**1.38**

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)  $1010111_2$

**1.39**

В системе счисления с некоторым основанием число 12 записывается в виде 110. Укажите это основание.

**1.40**

В системе счисления с некоторым основанием число 17 записывается в виде 101. Укажите это основание.

**1.41**

Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 17 оканчивается на 2.

**1.42**

Укажите через запятую в порядке возрастания все числа, не превосходящие 25, запись которых в двоичной системе счисления оканчивается на 101. Числа в ответе указывать в десятичной системе счисления.

**1.43**

Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 23 оканчивается на 2.

**1.2 Алгоритмизация и программирование**

*Алгоритмы, виды алгоритмов, описания алгоритмов. Формальное исполнение алгоритма.*

**1.44**

Цепочка из трех бусин формируется по следующему правилу: На первом месте в цепочке стоит одна из бусин А, Б, В. На втором – одна из бусин Б, В, Г. На третьем месте – одна из бусин А, В, Г, не стоящая в цепочке на первом или втором месте.

Какая из следующих цепочек создана по этому правилу?

- 1) АГБ      2) ВАГ      3) БГГ      4) ББГ



#### 1.45

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

#### 1.46

Для составления цепочек используются бусины, помеченные буквами А, В, С, D, E. На первом месте в цепочке стоит одна из бусин А, С, E. На втором – любая гласная, если первая буква согласная, и любая согласная, если первая гласная. На третьем месте – одна из бусин С, D, E, не стоящая в цепочке на первом месте.

Какая из перечисленных цепочек создана по этому правилу?

- 1) CBE                      2) ADD                      3) ECE                      4) EAD

#### 1.47

В понедельник в одном из классов должно быть проведено 4 урока - по математике, физике, информатике и биологии. Учителя высказали свои пожелания для составления расписания. Учитель математики хочет иметь первый или второй урок, учитель физики - второй или третий урок, учитель информатики - первый или четвертый, учитель биологии - третий или четвертый. Какой вариант расписания устроит всех учителей школы?

(Обозначения: М – математика, Ф – физика, И – информатика, Б – биология)

- 1) ИМБФ                      2) МФБИ                      3) МИФБ                      4) МБФИ

#### 1.48

Дешифровщику необходимо восстановить поврежденный фрагмент сообщения, состоящий из 4-х символов. Имеется достоверная информация, что использовано не более пяти букв (А, О, Б, В, Г), причем на втором месте стоит один из символов А, Г, О, В. На первом – одна из букв Б, В, Г, О, которой нет на третьем месте. На третьем месте – одна из букв О, А, Б, В, не стоящая в слове на втором месте. На четвертом месте – одна из букв А, Б, В, Г, которой не было на первом месте.

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

- 1) ВВАА                      2) БОБА                      3) ОВВА                      4) ГОАГ

#### 1.49

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*-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.

```
i := Длина (a)
k := 1
b := 'Т'
пока i > 1
нц
  c := Извлечь (a, i)
  b := Склеить (b, c)
  i := i - k
кц
```

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

- 1) 'КАРАТ'      2) 'ТРАК'      3) 'КРАТ'      4) 'ТКАРА'

### 1.50

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  $i$ -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,  $a := \text{'дом'}$ ).

Фрагмент алгоритма:

```
i := Длина (a)
k := 1
b := 'П'
пока i > 0
нц
  c := Извлечь (a, i)
  b := 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'

### 1.51

Имеется исполнитель Кузнечик, который живет на числовой оси. Система команд Кузнечика: «Вперед  $N$ » (Кузнечик прыгает вперед на  $N$  единиц), «Назад  $M$ » (Кузнечик прыгает назад на  $M$  единиц). Переменные  $N$  и  $M$  могут принимать любые целые положительные значения. Известно, что Кузнечик выполнил программу из 50 команд, в которой команд «Назад 2» на 12 больше, чем команд «Вперед 3». Других команд в программе не было. На какую одну команду можно заменить эту программу, чтобы Кузнечик оказался в той же точке, что и после выполнения программы?

### 1.52

У исполнителя Утроитель две команды, которым присвоены номера.

**1. вычти 1**

**2. умножь на 3**

Первая из них уменьшает число на экране на 1, вторая – увеличивает его в три раза.

Запишите порядок команд в программе получения из числа 3 числа 16, содержащей не более 5 команд, указывая лишь номера команд

(Например, программа **21211** это программа)

**умножь на 3**

**вычти 1**

**умножь на 3**

**вычти 1**

**вычти 1**

которая преобразует число 1 в 4 )

### 1.53

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.)

### 1.54

У исполнителя Квадратор две команды, которым присвоены номера:

**1. возведи в квадрат**

**2. прибавь 1**

Выполняя первую из них, Квадратор возводит число на экране в квадрат, а выполняя вторую, 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 )

### 1.55

Записано 6 строк, каждая имеет свой номер – от 0 до 5.

В нулевой строке записана цифра 0 (ноль).

Каждая последующая строка состоит из двух повторов предыдущей и добавленного в конец своего номера (в  $i$ -й строке в конце приписана цифра  $i$ ). Ниже показаны первые

четыре строки, сформированные по описанному правилу (в скобках записан номер строки):

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

Какая цифра стоит в последней строке на 62-м месте (считая слева направо)?

### 1.56

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)?

### 1.57

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

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

### 1.58

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

Черепашке был дан для исполнения следующий алгоритм

**Повтори 5 [Вперед 10 Направо 72]**

Какая фигура появится на экране?

- 1) Незамкнутая ломаная линия
- 2) Правильный треугольник
- 3) Квадрат
- 4) Правильный пятиугольник

### 1.59

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

(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 7 [Вперед 40 Направо  $n$ ],**

чтобы на экране появился правильный шестиугольник?

- 1) 30
- 2) 45
- 3) 50
- 4) 60

### 1.60

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.

**Налево  $m$** , вызывающая изменение направления движения на  $m$  градусов против часовой стрелки.

(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 13 [Вперед 12 Налево  $n$ ],**

чтобы на экране появился правильный двенадцатигульник?

- 1) 11
- 2) 30
- 3) 39
- 4) 60

### 1.61

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

3233241

Какую последовательность из трех 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?

### 1.62

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 (влево), 4 (вправо), переходя на соседнюю клетку в направлении, указанном в скобках. Если в этом направлении между клетками стоит стена, то Робот разрушается. Робот успешно выполнил программу

**1132432**

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

### 1.63

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

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

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

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

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

Цикл

ПОКА < условие > команда

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

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

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

НАЧАЛО

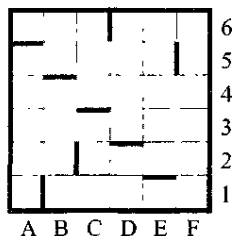
ПОКА < справа свободно > вправо

ПОКА < сверху свободно > вверх

ПОКА < слева свободно > влево

ПОКА < снизу свободно > вниз

КОНЕЦ



1) 1

2) 2

3) 3

4) 4

### 1.64

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)?

#### 1.65

Два игрока играют в следующую игру. Имеются три кучи камней, содержащих соответственно 2, 3, 4 камня. За один ход разрешается или удвоить количество камней в какой-нибудь куче, 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 куче становится  $\geq 15$  камней или во всех трех кучах суммарно становится  $\geq 25$  камней

Игроки ходят по очереди. Выяснить, кто выигрывает при правильной игре, - первый или второй игрок.

#### 1.66

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 куче, или добавляет 3 камня в какую-то кучу. Выигрывает игрок, после хода которого в одной из куч становится не менее 24 камней. Кто выигрывает при безошибочной игре обоих игроков – игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

#### 1.67

Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 4, а во второй – 3 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 куче или добавляет 2 камня в какую-то кучу. Выигрывает игрок, после хода которого общее число камней в двух кучах становится не менее 24 камней. Кто выигрывает при безошибочной игре обоих игроков – игрок, делающий первый ход или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

#### 1.68

Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 5, а во второй – 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или удваивает число камней в какой-то куче, или добавляет 4 камня в какую-то кучу. Выигрывает игрок, после хода которого в одной из куч становится не менее 22 камней. Кто выигрывает при безошибочной игре обоих игроков – игрок, делающий первый ход, или игрок, делающий второй ход? Как должен ходить выигрывающий игрок? Ответ обоснуйте.

#### 1.69

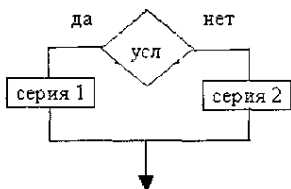
Два игрока играют в следующую игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. В начале игры фишка находится в точке с координатами (5,2). Ход состоит в том, что игрок перемещает фишку из точки с координатами (x,y) в одну из трех точек: или в точку с координатами (x+3,y), или в точку с координатами (x,y+3), или в

точку с координатами  $(x, y+4)$ . Выигрывает игрок, после хода которого расстояние от фишки до точки с координатами  $(0,0)$  не меньше 13 единиц. Кто выигрывает при безошибочной игре обоих игроков – игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте

**Основные алгоритмические конструкции: следование, ветвление, цикл. Блок-схемы. Переменные.**

**1.70**

Алгоритмическая конструкция какого типа изображена на фрагменте блок-схемы

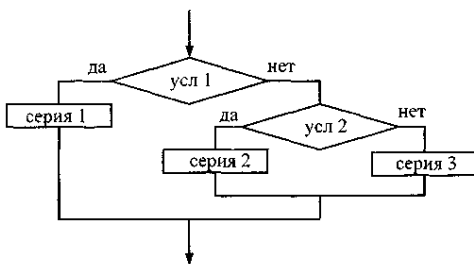


5

- 1) линейная
- 2) циклическая
- 3) разветвляющаяся
- 4) вспомогательная

**1.71**

Фрагмент блок-схемы



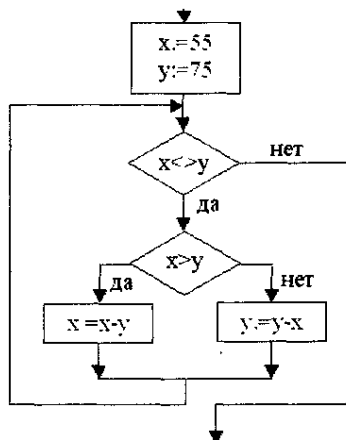
представляет алгоритм, который содержит две команды ветвления

- 1) команду ветвления в сокращенной форме, в которую вложена команда ветвления в полной форме
- 2) две команды ветвления в полной форме, одна из которой вложена в другую
- 3) две команды ветвления в сокращенной форме, одна из которой вложена в другую
- 4) команду ветвления в полной форме, в которую вложена команда ветвления в сокращенной форме



## 1.72

Определите значение целочисленной переменной  $x$  после выполнения следующего фрагмента программы.



1) 1

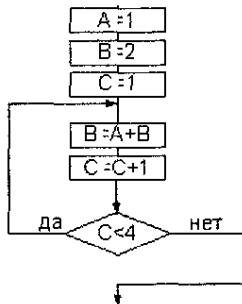
2) 5

3) 10

4) 15

## 1.73

Определите значение переменной  $B$  после выполнения следующего фрагмента алгоритма.



1) 6

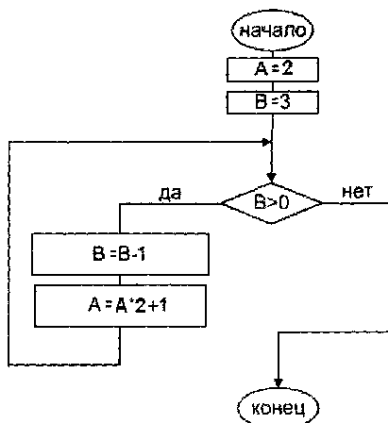
2) 5

3) 3

4) 4

1.74

Определите значение переменной A после выполнения следующего алгоритма:



1) 5

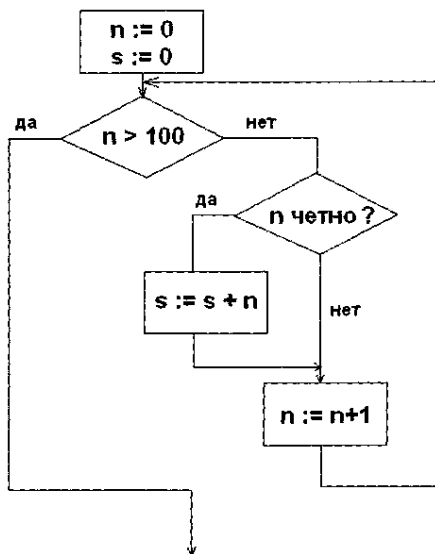
2) 11

3) 23

4) 47

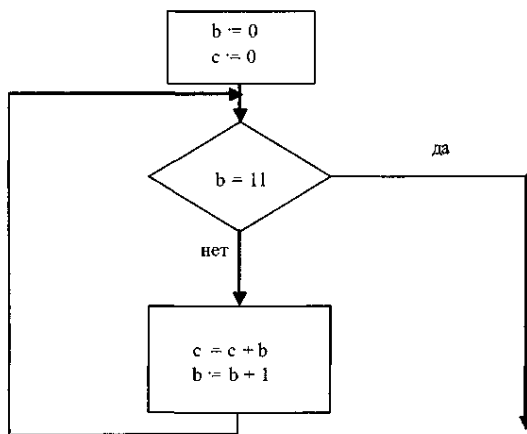
1.75

Определите значение переменной s после выполнения следующего фрагмента алгоритма:



## 1.76

Определите значение переменной  $c$  после выполнения фрагмента алгоритма:



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

1) 1

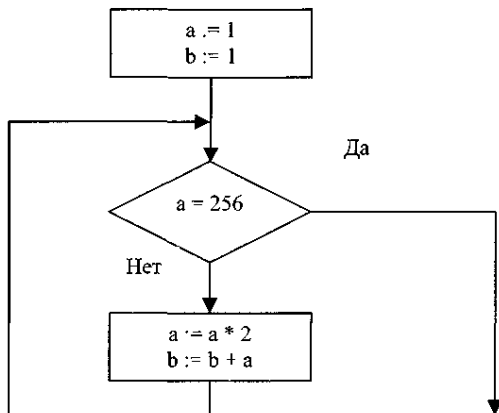
2) 45

3) 55

4) 66

## 1.77

Определите значение переменной  $b$  после выполнения фрагмента алгоритма:



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

1) 256

2) 511

3) 255

4) 512

## 1.78

Определите значение целочисленных переменных  $x$ ,  $y$  и  $t$  после выполнения фрагмента программы (ниже представлена одна и та же программа, записанная на разных языках программирования):

Бейсик	Паскаль	Алгоритмический
$x=5$ $y=7$ $t=x$ 'MOD - стандартная операция, вычисляющая остаток от деления нацело первого аргумента на второй $x= y \text{ MOD } x$ $y=t$	$x:=5;$ $y:=7;$ $t:=x;$ $x = y \bmod x;$ {mod - стандартная операция, вычисляющая остаток от деления нацело первого аргумента на второй} $y:=t;$	$x:=5,$ $y:=7;$ $t:=x;$ $x:=\text{mod}(y,x);$   mod - стандартная функция, вычисляющая остаток от деления нацело первого аргумента на второй  $y:=t,$

- 1)  $x=2, y=5, t=5$       2)  $x=7, y=5, t=5$       3)  $x=2, y=2, t=2$       4)  $x=5, y=5, t=5$

### 1.79

Определите значение целочисленных переменных  $a$  и  $b$  после выполнения фрагмента программы (ниже представлена одна и та же программа, записанная на разных языках программирования):

Бейсик	Паскаль	Алгоритмический
$a=42$ $b=14$ $a=a \setminus b$ $b=a * b$ $a=b \setminus a$ \ — стандартная операция, вычисляющая результат деления нацело первого аргумента на второй	$a:=42;$ $b:=14;$ $a:=a \text{ div } b;$ $b:=a * b;$ $a:=b \text{ div } a;$ {div — стандартная операция, вычисляющая результат деления нацело первого аргумента на второй}	$a:=42$ $b:=14$ $a:= \text{div}(a, b)$ $b:= a * b$ $a:= \text{div}(b, a)$   div — стандартная функция, вычисляющая результат деления нацело первого аргумента на второй

- 1)  $a = 42, b = 14$       2)  $a = 1, b = 42$       3)  $a = 0, b = 588$       4)  $a = 14, b = 42$

### 1.80

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	$a:=2468;$ $b:=(a \bmod 1000)*10;$ $a:=a \text{ div } 1000 + b;$ {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)  $a = 470, b = 468$

### 1.81

Определите значение целочисленных переменных  $x$ ,  $y$  и  $t$  после выполнения фрагмента программы:

Бейсик	Паскаль	Алгоритмический
$x=4$ $y=16$ $t=x$ ' MOD - стандартная операция, вычисляющая остаток от деления нацело первого аргумента на второй $x = y \text{ MOD } x$ $y = t + 1$	$x:=4$ ; $y:=16$ ; $t:=x$ ; $x:=y \bmod x$ , {mod - стандартная операция, вычисляющая остаток от деления нацело первого аргумента на второй} $y:=t+1$ ;	$x:=4$ ; $y:=16$ ; $t:=x$ ; $x:=\text{mod}(y,x)$ ;   mod - стандартная функция, вычисляющая остаток от деления нацело первого аргумента на второй  $y:=t+1$ ;

- 1)  $x=4, y=1, t=0$       2)  $x=0, y=5, t=4$       3)  $x=0, y=4, t=5$       4)  $x=4, y=1, t=0$

### 1.82

Определите значение целочисленных переменных  $b$  и  $c$  после выполнения фрагмента программы (ниже представлена одна и та же программа, записанная на разных языках программирования):

Бейсик	Паскаль	Алгоритмический
$a=37$ $b = a \text{ MOD } 10$ $c = a \setminus 10$ \ — стандартная операция, вычисляющая результат деления нацело первого аргумента на второй	$a:=37$ ; $b:=a \bmod 10$ ; $c:=a \text{ div } 10$ ; {div — стандартная операция, вычисляющая результат деления нацело первого аргумента на второй}	$a:=37$ $b:=\text{mod}(a,10)$ $c:=\text{div}(a,10)$ div — стандартная функция, вычисляющая результат деления нацело первого аргумента на второй

- 1)  $b = 3, c = 7$       2)  $b = 7, c = 3$       3)  $b = 3, c = 4$       4)  $b = 4, c = 3$

### 1.83

Определите значение целочисленных переменных  $a$  и  $b$  после выполнения фрагмента программы (ниже представлена одна и та же программа, записанная на разных языках программирования):

Бейсик	Паскаль	Алгоритмический
$a=20$ $b=7$ $a=a \setminus b$ $b=a * b$ $a=b \setminus a$ \ — стандартная операция, вычисляющая результат деления нацело первого аргумента на второй	$a:=20$ ; $b:=7$ ; $a:=a \text{ div } b$ ; $b:=a * b$ ; $a:=b \text{ div } a$ ; {div — стандартная операция, вычисляющая результат деления нацело первого аргумента на второй}	$a:=20$ $b:=7$ $a = \text{div}(a, b)$ $b := a * b$ $a = \text{div}(b, a)$ div — стандартная функция, вычисляющая результат деления нацело первого аргумента на второй

- 1)  $a = 7, b = 21$       2)  $a = 7, b = 7$       3)  $a = 7, b = 14$       4)  $a = 3, b = 21$

## 1.84

Определите значение переменной  $c$  после выполнения следующего фрагмента программы.

Бейсик	Паскаль	Алгоритмический
$a = 5$ $a = a + 6$ $b = -a$ $c = a - 2 * b$	$a := 5;$ $a := a + 6;$ $b := -a;$ $c := a - 2 * b;$	$a := 5$ $a := a + 6$ $b := -a$ $c := a - 2 * b$

1)  $c = -11$

2)  $c = 15$

3)  $c = 27$

4)  $c = 33$

**Работа с массивами**

## 1.85

Значения двумерного массива задаются с помощью вложенного оператора цикла в представленном фрагменте программы

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 5 FOR k=1 TO 5 B(n, k)=n+k NEXT k NEXT n	for n:=1 to 5 do for k:=1 to 5 do B[n,k]:=n+k;	<u>нц для n от 1 до 5</u> <u>нц для k от 1 до 5</u> B[n, k]:=n+k <u>кц</u> <u>кц</u>

Чему будет равно значение  $B(2,4)$ ?

1) 9

2) 8

3) 7

4) 6

## 1.86

Все элементы двумерного массива  $A$  размером  $10 \times 10$  элементов первоначально были равны 0. Затем значения элементов меняются с помощью вложенного оператора цикла в представленном фрагменте программы (ниже представлена одна и та же программа, записанная на разных языках программирования).

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 4 FOR k=n TO 4 A(n,k)=A(n,k)+1 A(k,n)=A(k,n)+1 NEXT k NEXT n	for n:=1 to 4 do for k:=n to 4 do begin A[n, k]:=A[n, k]+1; A[k, n]:=A[k, n]+1; end	<u>нц для n от 1 до 4</u> <u>нц для k от n до 4</u> A[n, k]:=A[n, k]+1 A[k, n]:=A[k, n]+1 <u>кц</u> <u>кц</u>

Сколько элементов массива в результате будут равны 1?

1) 0

2) 16

3) 12

4) 4

## 1.87

Все элементы двумерного массива  $A$  размером  $4 \times 4$  элемента первоначально были равны 0. Затем значения элементов меняются с помощью вложенного оператора цикла в представленном фрагменте программы (ниже представлена одна и та же программа, записанная на разных языках программирования).

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 4 FOR k=n TO 4 A(n,k)= 1 NEXT k NEXT n	for n =1 to 4 do for k =n to 4 do begin A[n, k]:= 1; end	нц для n от 1 до 4 нц для k от n до 4 A[n, k]:= 1 кц кц

Сколько элементов массива в результате будут равны 1?

### 1.88

Все элементы двумерного массива A размером 10x10 элементов первоначально были равны 1. Затем значения элементов меняются с помощью вложенного оператора цикла в представленном фрагменте программы.

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 4 FOR k=1 TO n+1 A(n,k)= A(n,k)-1 A(n,k+1)= A(n,k)-1 NEXT k NEXT n	for n:=1 to 4 do for k:=1 to n+1 begin A[n, k] :=A[n, k]-1; A[n, k+1]:=A[n, k]-1; end	нц для n от 1 до 4 нц для k от 1 до n+1 A[n, k]:=A[n, k]-1 A[n, k+1]:=A[n, k]-1 кц кц

Сколько элементов массива в результате будут равны 0?

### 1.89

1. Значения двумерного массива задаются с помощью вложенного оператора цикла в представленном фрагменте программы

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 500 FOR k=1 TO 500 B(n, k)=n*(n+1)*k/2 NEXT k NEXT n	for n = 1 to 500 do for k:=1 to 500 do B[n,k] := n*(n+1)*k/2; end	нц для n от 1 до 500 нц для k от 1 до 500 B[n, k] = n*(n+1)*k/2 кц кц

Чему будет равно значение B(19,21)?

### 1.90

Дан фрагмент программы, обрабатывающей двумерный массив A размера n×n

Бейсик	Паскаль	Алгоритмический
k = 1 FOR i = 1 TO n c = A(i,i) A(i,i) = A(k,i) A(k,i) = c NEXT i	k:=1; for i:=1 to n do begin c:=A[i,i]; A[i,i]:=A[k,i]; A[k,i]:=c end	k:=1 нц для i от 1 до n c:=A[i,i] A[i,i]:=A[k,i] A[k,i]:=c кц

Представим массив в виде квадратной таблицы, в которой для элемента массива A[i,j] величина i является номером строки, а величина j – номером столбца, в котором расположен элемент. Тогда данный алгоритм меняет местами

- 1) два столбца в таблице
- 2) две строки в таблице
- 3) элементы диагонали и k-ой строки таблицы
- 4) элементы диагонали и k-го столбца таблицы

### 1.91

Опишите на русском языке или одном из языков программирования алгоритм подсчета числа элементов равных максимальному в числовом массиве из 30 элементов.

### 1.92

Опишите на русском языке или одном 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).

### 1.93

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 нужно сосчитать сумму произведений первого и второго, третьего и четвертого, пятого и шестого элементов, и так далее.)

### 1.94

Опишите на русском языке или одном из языков программирования алгоритм вычисления разности между средним арифметическим максимального и минимального значений элементов заданного целочисленного массива из 30 элементов и средним арифметическим всех элементов этого массива.

## 1.3 Основы логики

### 1.95

Для какого имени истинно высказывание:

**¬(Первая буква имени гласная → Четвертая буква имени согласная)**

- |          |          |          |          |
|----------|----------|----------|----------|
| 1) ЕЛЕНА | 2) ВАДИМ | 3) АНТОН | 4) ФЕДОР |
|----------|----------|----------|----------|

### 1.96

Для какого имени истинно высказывание:

**Первая буква имени согласная ∧ (¬ Вторая буква имени согласная → Четвертая буква имени гласная)**

- |         |         |          |          |
|---------|---------|----------|----------|
| 1) ИВАН | 2) ПЕТР | 3) ПАВЕЛ | 4) ЕЛЕНА |
|---------|---------|----------|----------|

### 1.97

Для какого слова истинно высказывание

**(Вторая буква слова согласная ∨ Последняя буква слова гласная) → Первая буква слова гласная**

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



**1.98**

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

**1.99**

Для какого числа  $X$  истинно высказывание

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

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

**1.100**

Для какого числа  $X$  истинно высказывание

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

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

**1.101**

Какое логическое выражение равносильно выражению  $\neg(A \vee \neg B)$  ?

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

**1.102**

Укажите, какое логическое выражение равносильно выражению  $\neg(\neg A \wedge B)$

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

**1.103**

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

**1.104**

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

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

**1.105**

Упростите выражение  $\neg(\neg X \wedge \neg Y)$

**1.106**

Упростите выражение  $(\neg X \vee Y) \wedge X$

**1.107**

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

Дан фрагмент таблицы истинности выражения F.

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

Какое выражение соответствует F?

1)  $\neg X \wedge \neg Y \wedge Z$

2)  $\neg X \vee \neg Y \vee Z$

3)  $X \vee Y \vee \neg Z$

4)  $X \vee Y \vee Z$

### 1.108

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)  $X \vee Y \vee Z$

### 1.109

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

Дан фрагмент таблицы истинности выражения F.

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

Чему равно F?

1)  $X \wedge Y \wedge Z$

2)  $\neg X \vee \neg Y \vee Z$

3)  $X \wedge Y \wedge \neg Z$

4)  $\neg X \wedge \neg Y \wedge \neg Z$

### 1.110

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
1	0	0	1
0	0	0	1
1	1	1	0

Какое выражение соответствует F?

1)  $\neg X \wedge \neg Y \wedge \neg Z$

2)  $X \wedge Y \wedge Z$

3)  $X \vee Y \vee Z$

4)  $\neg X \vee \neg Y \vee \neg Z$

**1.111**

Укажите значения переменных K, L, M, N, при которых логическое выражение  $(K \vee M) \rightarrow (\neg L \vee M \vee N)$

**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  $K=0, L=1, M=0, N=1$ .

**1.112**

Сколько различных решений имеет уравнение

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

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 только количество таких наборов.

**1.113**

X, Y, Z - целые числа, для которых истинно высказывание

$$((Z < X) \vee (Z < Y)) \wedge \neg((Z+1) < X) \wedge \neg((Z+1) < Y)$$

Чему равно Z, если  $X=20, Y=10$ ?

**1.114**

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  $C=8, B=18$ ?

**1.115**

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  $A=45, C=43$ ?

**1.116**

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!»

Нadia: «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 высказываний истинно ровно три?

Ответ запишите в виде первой буквы имени

**1.117**

Три школьника, Миша (M), Коля (K) и Сергей (C), 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

#### 1.118

Мама, прибежавшая на звон разбившейся вазы, застала всех трех своих сыновей в совершенно невинных позах: Саша, Ваня и Коля делали вид, что происшедшее к ним не относится. Однако футбольный мяч среди осколков явно говорил об обратном.

- Кто это сделал? - спросила мама.

- Коля не бил по мячу, - сказал Саша. - Это сделал Ваня.

Ваня ответил: - Разбил Коля, Саша не играл в футбол дома. - Так я и знала, что вы друг на дружку сваливать будете, - рассердилась мама. - Ну, а ты что скажешь? - спросила она Колю. - Не сердись, мамочка! Я знаю, что Ваня не мог этого сделать. А я сегодня еще не сделал уроки, - сказал Коля. Оказалось, что один из мальчиков оба раза солгал, а двое в каждом из своих заявлений говорили правду.

Кто разбил вазу?

#### 1.119

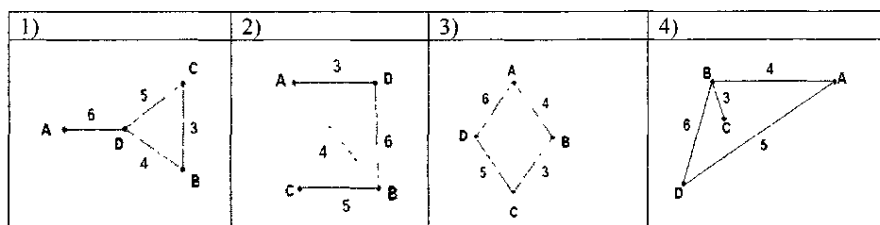
Классный руководитель пожаловался директору, что у него в классе появилась компания из 3-х учеников, один из которых всегда говорит правду, другой всегда лжет, а третий говорит через раз то ложь, то правду. Директор знает, что их зовут Коля, Саша и Миша, но не знает, кто из них правдив, а кто – нет. Однажды все трое прогуляли урок астрономии. Директор знает, что никогда раньше никто из них не прогуливал астрономию. Он вызвал всех троих в кабинет и поговорил с мальчиками. Коля сказал: "Я всегда прогуливаю астрономию. Не верьте тому, что скажет Саша". Саша сказал: "Это был мой первый прогул этого предмета". Миша сказал: "Все, что говорит Коля, – правда". Директор понял, кто из них кто. Расположите первые буквы имен мальчиков в порядке: "говорит всегда правду", "всегда лжет", "говорит правду через раз". (Пример: если бы имена мальчиков были Рома, Толя и Вася, ответ мог бы быть РТВ)

#### 1.4. Моделирование

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

#### 1.120

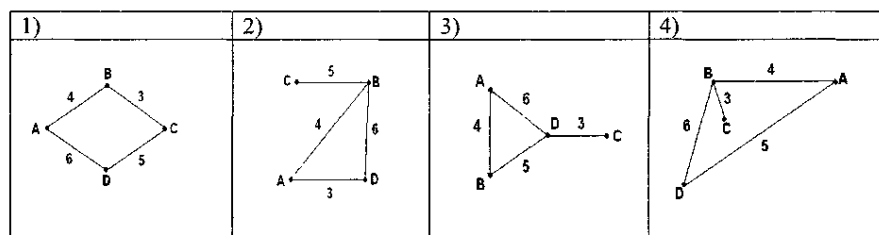
	A	B	C	D
A		4		6
B	4		3	
C		3		5
D	6		5	



### 1.121

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

	A	B	C	D
A		4		5
B	4		3	6
C		3		
D	5	6		



### 1.122

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

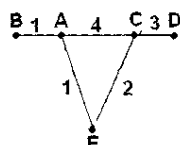
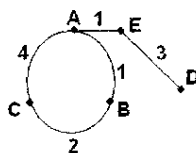
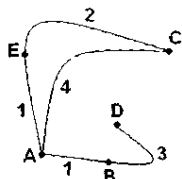
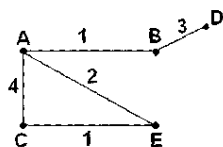
	A	B	C	D	E
A		1	4		1
B	1			3	
C	4				2
D		3			
E	1		2		

1)

2)

3)

4)



### 1.123

Таблица стоимости перевозок устроена следующим образом: числа, стоящие на пересечениях строк и столбцов таблицы 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)	2)	3)	4)																																																																																																																																																
<table><tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr><tr><td>A</td><td></td><td></td><td>3</td><td>1</td><td></td></tr><tr><td>B</td><td></td><td></td><td>4</td><td></td><td>2</td></tr><tr><td>C</td><td>3</td><td>4</td><td></td><td></td><td>2</td></tr><tr><td>D</td><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>E</td><td></td><td>2</td><td>2</td><td></td><td></td></tr></table>		A	B	C	D	E	A			3	1		B			4		2	C	3	4			2	D	1					E		2	2			<table><tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr><tr><td>A</td><td></td><td></td><td>3</td><td>1</td><td>1</td></tr><tr><td>B</td><td></td><td></td><td>4</td><td></td><td></td></tr><tr><td>C</td><td>3</td><td>4</td><td></td><td></td><td>2</td></tr><tr><td>D</td><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>E</td><td>1</td><td></td><td>2</td><td></td><td></td></tr></table>		A	B	C	D	E	A			3	1	1	B			4			C	3	4			2	D	1					E	1		2			<table><tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr><tr><td>A</td><td></td><td></td><td>3</td><td>1</td><td></td></tr><tr><td>B</td><td></td><td></td><td>4</td><td></td><td>1</td></tr><tr><td>C</td><td>3</td><td>4</td><td></td><td></td><td>2</td></tr><tr><td>D</td><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>E</td><td></td><td>1</td><td>2</td><td></td><td></td></tr></table>		A	B	C	D	E	A			3	1		B			4		1	C	3	4			2	D	1					E		1	2			<table><tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr><tr><td>A</td><td></td><td></td><td></td><td>1</td><td></td></tr><tr><td>B</td><td></td><td></td><td>4</td><td></td><td>1</td></tr><tr><td>C</td><td></td><td>4</td><td></td><td>4</td><td>2</td></tr><tr><td>D</td><td>1</td><td></td><td>4</td><td></td><td></td></tr><tr><td>E</td><td></td><td>1</td><td>2</td><td></td><td></td></tr></table>		A	B	C	D	E	A				1		B			4		1	C		4		4	2	D	1		4			E		1	2		
	A	B	C	D	E																																																																																																																																														
A			3	1																																																																																																																																															
B			4		2																																																																																																																																														
C	3	4			2																																																																																																																																														
D	1																																																																																																																																																		
E		2	2																																																																																																																																																
	A	B	C	D	E																																																																																																																																														
A			3	1	1																																																																																																																																														
B			4																																																																																																																																																
C	3	4			2																																																																																																																																														
D	1																																																																																																																																																		
E	1		2																																																																																																																																																
	A	B	C	D	E																																																																																																																																														
A			3	1																																																																																																																																															
B			4		1																																																																																																																																														
C	3	4			2																																																																																																																																														
D	1																																																																																																																																																		
E		1	2																																																																																																																																																
	A	B	C	D	E																																																																																																																																														
A				1																																																																																																																																															
B			4		1																																																																																																																																														
C		4		4	2																																																																																																																																														
D	1		4																																																																																																																																																
E		1	2																																																																																																																																																

### 1.124

Между четырьмя крупными аэропортами, обозначенными кодами BOK, EDT, LAA и SAK, ежедневно выполняются авиарейсы. Приведён фрагмент расписания перелётов между этими аэропортами:

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
BOK	EDT	03:10	05:15
SAK	EDT	03:35	05:40
SAK	BOK	07:30	11:30
LAA	BOK	10:15	12:45
BOK	LAA	10:40	15:05
SAK	LAA	11:20	13:25
BOK	SAK	13:35	17:25
EDT	BOK	14:20	16:20
LAA	SAK	14:50	16:30
EDT	SAK	15:55	18:10

Путешественник находится в аэропорту BOK в полночь (0 00). Определите самое раннее время, когда он может оказаться в аэропорту SAK

- 1) 11:30                      2) 16 30                      3) 17 25                      4) 18:10

## 2. Информационные и коммуникационные технологии

### 2.1. Файловые системы

#### 2.1

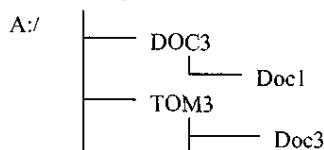
В некотором каталоге хранился файл **Дневник.txt**. После того, как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл **Дневник.txt**, полное имя файла стало

**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

#### 2.2

Дано дерево каталогов.



Определите полное имя файла Doc3

- 1) A:/DOC3  
2) A /DOC3/Doc3  
3) A:/ DOC3/Doc1  
4) A:/TOM3/Doc3

#### 2.3

В некотором каталоге хранился файл **txt.doc**. После того, как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл **txt.doc**, полное имя файла стало **P:\doc\txt\letter\txt.doc**. Каково было полное имя этого файла до перемещения?

#### 2.4

Ученик работал в каталоге **A:\Школа\8\Петров**. Учитель сказал ему: «Перейдите в дереве каталогов на уровень выше, спуститесь в подкаталог **Физика** и откройте файл **Урок12**». Каково полное имя файла, который должен был открыть ученик?

- 1) A.\Школа\8\Физика\Урок12  
2) A \Школа\Физика\8\Урок12  
3) A:\Школа\Физика\Петров\Урок12  
4) A \Школа\8\Физика\Петров\Урок12

## 2.5

В некотором каталоге хранился файл **Задача5**. После того, как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл **Задача5**, полное имя файла стало **Е:\Класс9\Физика\Задачник\Задача5**. Каково было полное имя этого файла до перемещения?

- 1) Е:\Физика\Задачник\Задача5
- 2) Е:\Физика\Задача5
- 3) Е:\Класс9\Задачник\Задача5
- 4) Е:\Класс9\Физика\Задача5

## 2.6

Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

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

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

Определите, какое из указанных имен файлов удовлетворяет маске:

**?ba\*r.txt**

- |            |             |            |             |
|------------|-------------|------------|-------------|
| 1) bar.txt | 2) obar.txt | 3) obar.xt | 4) bart.txt |
|------------|-------------|------------|-------------|

## 2.2. Обработка графической информации

### 2.7

Для хранения растрового изображения размером 64х64 пикселя отвели 1,5 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

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

### 2.8

Для хранения растрового изображения размером 128х128 пикселей отвели 4 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

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

### 2.9

Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 64х64 пикселя, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно

- |        |      |        |      |
|--------|------|--------|------|
| 1) 128 | 2) 2 | 3) 256 | 4) 4 |
|--------|------|--------|------|

### 2.10

Для хранения растрового изображения размером 64х32 пикселя отвели 1 килобайт памяти. Каково максимально возможное число цветов в палитре изображения?

- |       |       |       |         |
|-------|-------|-------|---------|
| 1) 16 | 2) 32 | 3) 64 | 4) 1024 |
|-------|-------|-------|---------|

### 2.11

Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 256х256 пикселей, если известно, что в изображении используется палитра из  $2^{16}$  цветов. Саму палитру хранить не нужно

- |        |        |         |         |
|--------|--------|---------|---------|
| 1) 128 | 2) 512 | 3) 1024 | 4) 2048 |
|--------|--------|---------|---------|



## 2.12

Каков минимальный объем памяти (в байтах), достаточный для хранения любого черно-белого растрового изображения размером 32х32 пикселя, если известно, что в изображении используется не более 16 градаций серого цвета

## 2.13

Для хранения растрового изображения размером 128х128 пикселей отвели 2 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

## 2.14

Для кодирования цвета фона страницы Интернет используется атрибут bgcolor="#XXXXXX", где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. Какой цвет будет у страницы, заданной тэгом <body bgcolor="#000000">?

- 1) черный                      2) зеленый                      3) красный                      4) синий

## 2.3. Обработка информации в электронных таблицах

*Работа с формулами, абсолютные и относительные ссылки*

### 2.15

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

у после того, как ячейку A1 скопируют в ячейку B1?

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

### 2.17

В ячейке C2 электронной таблицы записана формула =B3+2\*\$D2. Какой вид приобретет формула после того, как ячейку C2 скопируют в ячейку B1?

### 2.18

В ячейке C2 записана формула =\$E\$3+D2. Какой вид приобретет формула, после того как ячейку C2 скопируют в ячейку B1?

*Примечание* знак \$ используется для обозначения абсолютной адресации.

- 1) =\$E\$3+C1  
2) =\$D\$3+D2  
3) =\$E\$3+E3  
4) =\$F\$4+D2

### 2.19

Дан фрагмент электронной таблицы:

	A	B	C	D
1	5	2	4	
2	10	1	6	

В ячейку D2 введена формула =A2\*B1+C1

В результате в ячейке D2 появится значение:

- 1) 6                      2) 14                      3) 16                      4) 24

## 2.20

Дан фрагмент электронной таблицы:

	A	B	C	D
1	6	=A1/3	=A1-B1	=B2+C1
2	=C1+1	1	6	

Найдите числовое значение ячейки D1.

## 2.21

В электронной таблице значение формулы =СУММ(B1:B2) равно -5. Чему равно значение ячейки B3, если значение формулы =СРЗНАЧ(B1..B3) равно 2?

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

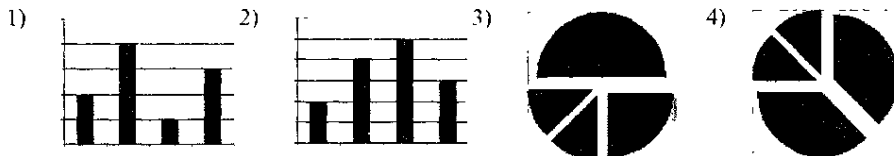
### Диаграммы и графики

## 2.22

Дан фрагмент электронной таблицы:

	A	B
1	=A4-B1	1
2	=A1+2	2
3	=A2+B1	
4	=B1+B2	

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



## 2.23

Дан фрагмент электронной таблицы:

	A	B	C	D
1		7	5	
2	=(B1-C1)/2	=C1-4	=B2+A2	=C1-B2

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



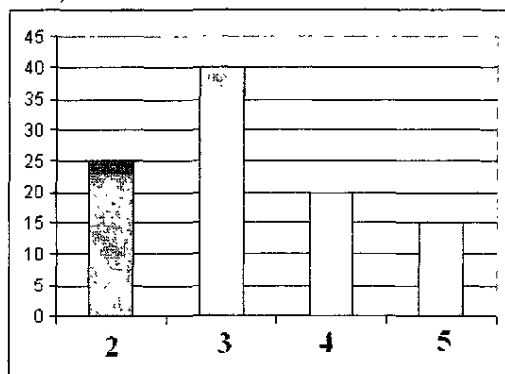
## 2.24

Дан фрагмент электронной таблицы:

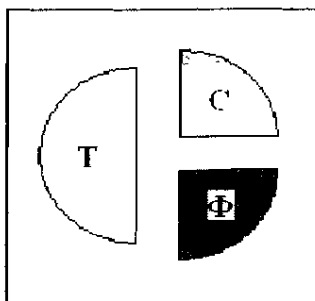
В цехе трудятся рабочие трех специальностей – токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд не меньший второго и не больший пятого. На диаграмме I отражено количество рабочих с различными разрядами, а на диаграмме II – распределение рабочих по специальностям.

Каждый рабочий имеет только одну специальность и один разряд.

I)



II)



Имеются четыре утверждения.

А Все фрезеровщики могут быть третьего разряда

Б) Все токари могут быть пятого 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

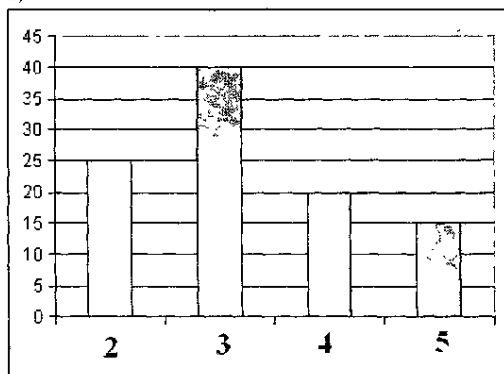
## 2.25

Дан фрагмент электронной таблицы:

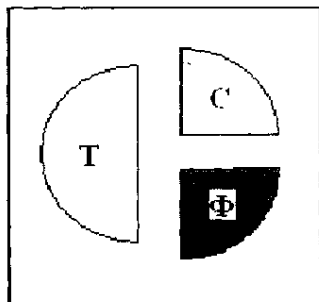
В цехе трудятся рабочие трех специальностей – токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд не меньший второго и не больший пятого. На диаграмме I отражено количество рабочих с различными разрядами, а на диаграмме II – распределение рабочих по специальностям.

Каждый рабочий имеет только одну специальность и один разряд.

I)



II)



Имеются четыре утверждения.

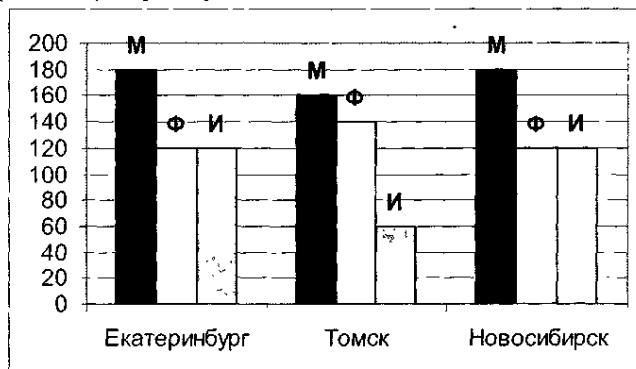
- А) Все рабочие третьего разряда могут быть токарями  
 Б) Все рабочие третьего разряда могут быть фрезеровщиками  
 В) Все фрезеровщики могут быть пятого разряда  
 Г) Все токари могут быть четвертого разряда

Какое из этих утверждений следует из анализа обеих диаграмм?

- 1) А                      2) Б                      3) В                      4) Г

## 2.26

На диаграмме показано количество призеров олимпиады по информатике (И), математике (М), физике (Ф) в трех городах России.



Какая из диаграмм правильно отражает соотношение призеров из всех городов по каждому предмету?

- 1) 2) 3) 4)

## 2.4. Базы данных

### 2.27

Ниже 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
Сергеевко	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

### 2.28

Для каждого файла в таблицу записывался исходный размер файла (поле РАЗМЕР), а также размеры архивов, полученных после применения к файлу различных архиваторов: программы WinZIP (поле ZIP), программы WinRAR (поле RAR) и программы StuffIt (поле SIT) Вот начало этой таблицы (все размеры в таблице - в килобайтах):

Имя файла	РАЗМЕР	ZIP	RAR	SIT
Аквариум.mw2	296	124	88	92
Муар.mw2	932	24	20	28

Нужно отобрать файлы, исходный размер которых больше 1 мегабайта и размер которых при использовании WinRAR уменьшился более чем в 4 раза. Для этого достаточно найти в таблице записи, удовлетворяющие условию:

- 1) (РАЗМЕР > 1000) ИЛИ (РАЗМЕР / RAR > 4)  
2) (РАЗМЕР > 1024) И (RAR < 256)  
3) (РАЗМЕР > 1024) И (РАЗМЕР / RAR > 4)  
4) (РАЗМЕР > 1024) ИЛИ (РАЗМЕР / RAR > 4)

### 2.29

Для каждого файла в таблицу записывался исходный размер файла (поле РАЗМЕР), а также размеры архивов, полученных после применения к файлу различных архиваторов: программы WinZIP (поле ZIP), программы WinRAR (поле RAR) и программы StuffIt (поле SIT) Вот начало этой таблицы (все размеры в таблице - в килобайтах):

Имя файла	РАЗМЕР	ZIP	RAR	SIT
Аквариум mw2	296	124	88	92
Муар.mw2	932	24	20	28

Нужно отобрать файлы, исходный размер которых больше 1 мегабайта и размер которых при использовании WinZip уменьшился более чем в 2 раза. Для этого достаточно найти в таблице записи, удовлетворяющие условию:

- 1) (РАЗМЕР > 100) ИЛИ (РАЗМЕР / ZIP > 2)
- 2) (РАЗМЕР > 100) И (ZIP < 50)
- 3) (РАЗМЕР > 1024) И (РАЗМЕР / ZIP > 2)
- 4) (РАЗМЕР > 1024) ИЛИ (РАЗМЕР / ZIP > 2)

### 2.30

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

### 2.31

Результаты тестирования представлены в таблице:

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 ней удовлетворяют условию  
«Пол='ж' ИЛИ Химия>Биология»?

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

### 2.32

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
Гордизани 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) Виктор М. П.   | 3) Михальчук М.С |
| 2) Гордизани Б. Ш | 4) Шапсугов М.Х. |

## 2.5. Телекоммуникационные технологии

### Адресация в сети Интернет

#### 2.33

Идентификатор некоторого ресурса сети Интернет имеет следующий вид:  
**http://www.ftp.ru/index.html**

Какая часть этого идентификатора указывает на протокол, используемый для передачи ресурса?

- |        |        |         |         |
|--------|--------|---------|---------|
| 1) www | 2) ftp | 3) http | 4) html |
|--------|--------|---------|---------|

#### 2.34

Идентификатор некоторого ресурса сети Интернет: **ftp://home.net/www.doc**. Какая часть этого идентификатора является именем сервера, на котором расположен ресурс?

#### 2.35

Доступ к файлу **net.edu**, находящемуся на сервере **ru.com** осуществляется по протоколу **ftp**.

В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла.

А	ftp
Б	ru
В	://
Г	edu
Д	com
Е	net
Ж	/

#### 2.36

Доступ к файлу **text.net**, находящемуся на сервере **www.ru** осуществляется по протоколу **http**. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла

А	text
Б	://
В	/
Г	www
Д	http
Е	.net
Ж	.ru

### 2.37

Доступ к файлу `www.tx1`, 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	.txt
Б	http
В	/
Г	./
Д	net
Е	www
Ж	ftp

### 2.38

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

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

2.131	20	.31	2.19
А	Б	В	Г

### Поиск информации в сети Интернет

#### 2.39

Каким условием нужно воспользоваться для поиска в сети Интернет информации о цветах, растущих на островах Тайвань или Хонсю (для обозначения логической операции «ИЛИ» используется символ `|`, а для операции «И» - символ `&`)?

- 1) цветы`&`(Тайвань|Хонсю)
- 2) цветы`&`Тайвань`&`Хонсю
- 3) цветы|Тайвань|Хонсю
- 4) цветы`&`(остров|Тайвань|Хонсю)

#### 2.40

В сети Интернет требуется найти информацию о большой белой акуле (другое распространенное название этой акулы – кархародон). Сформулируйте запрос к поисковой системе, учитывая, что для обозначения логической операции «ИЛИ» в запросе используется символ `|`, а для логической операции «И» - `&`.

#### 2.41

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

Для обозначения логической операции «ИЛИ» в запросе используется символ `|`, а для логической операции «И» - `&`.



А	законы & физика
Б	законы   (физика & биология)
В	законы & физика & биология & химия
Г	законы   физика   биология

## 2.42

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

## 2.43

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

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

1	принтеры & сканеры & продажа
2	принтеры & продажа
3	принтеры   продажа
4	принтеры   сканеры   продажа

## 2.6. Технологии программирования

### 2.44

В программе вводятся с клавиатуры три различных целых числа в переменные **a**, **b**, **c**. Далее в программе требуется преобразовать исходные данные так, чтобы в переменной **a** оказалось наименьшее число, в переменной **b** оказалось среднее число, в переменной **c** – наибольшее число.

Программа на Паскале

```
var
  a, b, c, x. integer;
begin
  write('Введите a,b,c'); read(a,b,c);
  if a>b then begin x:=a; a:=b; b:=x end;
  if b>c then begin x:=b; b:=c; c:=x end
end.
```

Программа на Бейсике

```
PRINT " Введите a,b,c"
INPUT a,b,c
IF a>b THEN x=a: a=b: b=x
IF b>c THEN x=b: b=c: c=x
```

Программист допустил ошибку при написании программы

1) Привести пример таких исходных значений **a**, **b**, **c**, после обработки которых исходной программой соотношение  $a < b < c$  не будет выполнено.

2) Исправить ошибку в программе, так чтобы она полностью соответствовала постановке задачи

#### 2.45

Рассматривается стандартная шахматная доска размером 8x8. Примем, что  $i$  – номер вертикали (может принимать значения от 1 до 8),  $j$  – номер горизонтали (также может принимать значения от 1 до 8). В левом нижнем углу, т.е. на поле  $i=1, j=1$  (это поле черного цвета) стоит черный король. Напомним, что король может ходить на 1 клетку в любом направлении (по горизонтали, вертикали или диагонали). В правом нижнем углу, т.е. на поле  $i=8, j=1$  (это поле белого цвета) стоит белый король. Введены обозначения:  $P(i,j)$  – минимальное число ходов, за которое черный король может попасть на поле  $(i,j)$ ,  $V(i,j)$  – минимальное число ходов, за которое белый король может попасть на поле  $(i,j)$ .

Программист написал программу, в которой требовалось определить все такие поля  $(i,j)$ , для которых  $P(i,j) = V(i,j)$ , и выдать на экран соответствующие значения  $i,j$  (текст программы приведен ниже)

1) Выдаст ли программа, написанная программистом, поле, для которого  $i=4, j=5$ ?

2) Указать все из перечисленных ниже полей, которые удовлетворяют постановке задачи, т.е. для таких полей должно быть выполнено  $P(i,j) = V(i,j)$

$(i=1, j=8), (i=2, j=8), (i=1, j=7), (i=5, j=5), (i=8, j=6)$

3) Видно, что программист допустил ошибку в программе. Укажите, какую доработку программы нужно провести, чтобы она соответствовала постановке задачи (такая доработка может быть проведена неединственным образом – годится любой правильный вариант доработки)

Программа на языке Паскаль	Программа на языке Бейсик
<pre>VAR i,j: integer; BEGIN writeln('искомые поля'), for j=5 to 8 do for i=1 to 8 do begin if (i=9-j) OR (i=j) then writeln('i=',i, 'j=',j); end; END.</pre>	<pre>PRINT "Искомые поля" FOR J=5 TO 8 FOR I=1 TO 8 IF (I=9-J) OR (I=J) THEN PRINT "I="; I PRINT "J="; J ENDIF NEXT I NEXT J END</pre>

#### 2.46

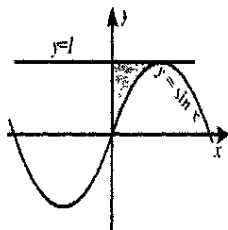
Требовалось написать программу, которая решает неравенство « $ax+b>0$ » относительно  $x$  для любых чисел  $a$  и  $b$ , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre> var a,b,x: real; begin   readln(a,b,x);   if a = 0 then     write('любое число')   else     if a &gt; 0 then       write('x &gt;', -b/a)     else       write('x &lt;', -b/a)     end   end end </pre>	<pre> INPUT a, b, x IF a = 0 THEN   PRINT "любое число" ELSE   IF a &gt; 0 THEN     PRINT "x &gt; ", -b/a   ELSE     PRINT "x &lt; ", -b/a   END END </pre>	<pre> void main(void) {   float a,b,x;   scanf("%f%f%f", &amp;a,&amp;b,&amp;x);   if (a==0)     printf("любое число");   else     if (a&gt;0)       printf("x&gt;=%f", -b/a);     else       printf("x&lt;=%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 нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы)

2.47



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

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre> var x,y: real; begin   readln(x,y);   if y&lt;=1 then     if x&gt;=0 then       if y&gt;=sin(x) then         write('принадлежит')       else         write('не принадлежит')       end     end   end end. </pre>	<pre> INPUT x, y IF y&lt;=1 THEN   IF x&gt;=0 THEN     IF y&gt;=SIN(x) THEN       PRINT "принадлежит"     ELSE       PRINT "не принадлежит"     ENDIF   ENDIF ENDIF END </pre>	<pre> void main(void) {   float x,y;   scanf("%f%f",&amp;x,&amp;y);   if (y&lt;=1)     if (x&gt;=0)       if (y&gt;=sin(x))         printf("принадлежит");       else         printf("не принадлежит");     } } </pre>

Последовательно выполните следующее:

- 1) Приведите пример таких чисел  $x$ ,  $y$ , при которых программа неверно решает поставленную задачу.

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

#### 2.48

На вход программе подаются 366 строк, которые содержат информацию о среднесуточной температуре всех дней 2004 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 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 программу на языке Паскаль или Бейсик, которая будет выводить на экран информацию о месяце (месяцах) среднемесячная температура у которого (которых) наименее отклоняется от среднегодовой. В первой строке вывести среднегодовую температуру. Найденные значения для каждого из месяцев следует выводить в отдельной строке в виде: номер месяца, значение среднемесячной температуры, отклонение от среднегодовой температуры.

#### 2.49

Во входном файле `meteo.dat` 365 строк, которые содержат информацию о среднесуточной температуре всех дней 2003 г. 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 записано значение температуры — число со знаком плюс или минус, с точностью до 1 цифры после десятичной точки. Данная информация отсортирована по значению температуры, т.е. хронологический порядок нарушен. Требуется написать программу на языке Паскаль или Бейсик, которая рассчитает и выведет на экран следующую выходную информацию:

- 1) среднегодовую температуру (в отдельной строчке вывода),
- 2) информацию о месяцах, для которых абсолютная величина разности среднемесячной и среднегодовой температуры не превосходит 5 градусов. Значения для каждого из месяцев, удовлетворяющих этому условию, следует выводить в отдельной строке в виде: номер месяца, значение среднемесячной температуры.

#### 2.50

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 на языке Паскаль или Бейсик, 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:

`batai.`

В данном случае программа должна вывести

`aib`

#### 2.51

На вход программе подаются сведения о пассажирах, сдавших свой багаж в камеру хранения. В первой строке задано текущее время: через двоеточие два целых числа, соот-

ветствующие часам (от 00 до 23 – ровно 2 символа) и минутам (от 00 до 59 – ровно 2 символа). Во второй строке сообщается количество пассажиров N, которое не меньше 10, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат:

<Фамилия> <время освобождения ячейки>,

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

Требуется написать на языке Паскаль или Бейсик программу, выводющую фамилии пассажиров, которые в ближайшие 2 часа должны освободить ячейки, в хронологическом порядке освобождения ячеек.

Пример входных данных:

10 00

3

Иванов 12:00

Петров 10:00

Сидоров 12:12

Результат работы программы для этого примера

Петров

Иванов

## 2.52

На вход программе подаются сведения о номерах школ учащихся, участвовавших в олимпиаде. В первой строке сообщается количество учащихся N, каждая из следующих N строк имеет формат: <Фамилия> <Инициалы> <номер школы>, где <Фамилия> – строка, состоящая не более чем из 20 символов, <Инициалы> – строка, состоящая из 4-х символов (буква, точка, буква, точка), <номер школы> – не более чем двузначный номер. <Фамилия> и <Инициалы>, а также <Инициалы> и <номер школы> разделены одним пробелом. Пример входной строки:

Иванов П.С. 57

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить на экран информацию, из каких школ было меньше всего участников олимпиады (но из этих школ был хотя бы один участник).

# ТРЕНИРОВОЧНЫЕ ВАРИАНТЫ 2009 г.

## Вариант 1

### Часть 1

При выполнении заданий этой части в бланке ответов № 1 под номером выполняемого вами задания (A1 – A20) поставьте знак «×» в клеточку, номер которой соответствует номеру выбранного вами ответа.

**A1**

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 720 бит. Какова длина сообщения в символах?

- 1) 90                      2) 45                      3) 180                      4) 720

**A2**

В велокроссе участвуют 779 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 280 велосипедистов?

- 1) 280 бит                      2) 779 бит                      3) 280 байт                      4) 350 байт

**A3**

Дано  $a=92_{16}$ ,  $b=224_8$ . Какое из чисел  $c$ , записанных в двоичной системе, отвечает условию  $a < c < b$ ?

- 1) 10010011                      2) 10001110                      3) 10001010                      4) 10001100

**A4**

Чему равна сумма чисел  $41_8$  и  $57_{16}$ ?

- 1)  $120_8$                       2)  $170_8$                       3)  $68_{16}$                       4)  $1000000_2$

**A5**

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

Бейсик	Паскаль	Алгоритмический
$a = 8 + 2 * 5$ $b = (a \text{ MOD } 10) + 14$ $a = (b \setminus 10) + 3$ $\setminus$ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a := 8 + 2 * 5;$ $b := (a \bmod 10) + 14;$ $a := (b \div 10) + 3$ $\{ \div \text{ и } \bmod \text{ — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно} \}$	$a := 8 + 2 * 5$ $b := \text{mod}(a, 10) + 14$ $a := \text{div}(b, 10) + 3$ $\{ \div \text{ и } \text{mod} \text{ — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно} \}$

- 1)  $a = 4$ ,  $b = 14$   
 2)  $a = 4$ ,  $b = 15$   
 3)  $a = 5$ ,  $b = 22$   
 4)  $a = 5$ ,  $b = 23$

\*Тренировочные варианты составлены в соответствии с кодификатором элементов содержания и демонстрационным вариантом ЕГЭ 2009 г.  
 Кодификатор и демоверсия 2009 г. см. на сайте [www.fipi.ru](http://www.fipi.ru).

**A6** Дан фрагмент программы, обрабатывающей массив A из n элементов:

Бейсик	Паскаль	Алгоритмический
<pre> j = 1 FOR i = 1 TO n IF A(i) &gt; A(j) THEN j = i NEXT i s = j         </pre>	<pre> j:=1; for i:=1 to n do begin     if A[i]&gt;A[j] then         j:=i end s:=j         </pre>	<pre> j:=1 нц для i от 1 до n     если A[i]&gt;A[j] то         j:=i     все кц s:=j         </pre>

Чему будет равно значение переменной s после выполнения данного алгоритма, при любых значениях элементов массива A?

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

**A7** Для какого из указанных значений X истинно высказывание

$\neg((X>2) \rightarrow (X>3))$ ?

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

**A8** Укажите, какое логическое выражение равносильно выражению  $A \wedge \neg(B \vee \neg C)$

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

**A9** 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
1	0	0	0
0	0	1	1
0	1	0	0

Какое выражение соответствует F?

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

**A10**

Между четырьмя крупными аэропортами, обозначенными кодами DLU, IGT, OPK и QLO, ежедневно выполняются авиарейсы. Приведён фрагмент расписания перелётов между этими аэропортами:

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
QLO	IGT	06:20	08:35
IGT	DLU	10:25	12:35
DLU	IGT	11:45	12:30
OPK	QLO	12:15	14:25
QLO	DLU	12:45	16:35
IGT	QLO	13:45	15:40
DLU	QLO	13:40	17:25
DLU	OPK	15:30	17:15
QLO	OPK	17:35	19:30
OPK	DLU	19:40	21:55

Путешественник находится в аэропорту DLU в полночь (0:00). Определите самое раннее время, когда он может оказаться в аэропорту QLO.

- 1) 15:40                      2) 16:35                      3) 17:15                      4) 17:25

**A11**

Для кодирования букв А, Б, В, Г решили использовать двухразрядные последовательные двоичные числа (от 00 до 11 соответственно). Если таким способом закодировать последовательность символов ВАГБ и записать результат шестнадцатеричным кодом, то получится:

- 1) 8D                      2) CADB                      3) 813                      4) 2031

**A12**

Для составления цепочек используются бусины, помеченные буквами М, Н, О, Р, Е. Цепочка формируется по следующему правилу: в середине цепочки стоит одна из бусин М, О, Е. На третьем месте – бусина, помеченная любой гласной, если первая бусина – согласной, или любой согласной, если первая – гласной. На первом месте – одна из бусин О, Р, Е, не стоящая в цепочке посередине. Какая из перечисленных цепочек создана по этому правилу?

- 1) OPN                      2) PEE                      3) POM                      4) EEO

**A13**

Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «\*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «\*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **hel?lo.c?\***.

- 1) hel!lo.cpp                      2) hel!lo c                      3) helholo.cpp                      4) helolo c



A14

Это таблица, в которой для каждого графического файла в поле «Глубина» указана глубина цвета (количество бит на пиксель) и в полях «BMP», «GIF» и «JPEG» указан размер (в килобайтах) при сохранении этого файла в соответствующих форматах.

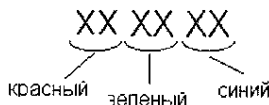
Имя файла	Глубина	BMP	GIF	JPEG
photo	24	768	552	226
mypicture	16	234	93	129
chaos	24	103	100	41
newpicture	16	234	45	89
screen	24	900	124	217

Таблицу отсортировали по убыванию значения выражения GIF/BMP. Какой файл находится в первой строке после сортировки?

- 1) photo
- 2) newpicture
- 3) screen
- 4) chaos

A15

Для кодирования цвета фона страницы Интернет используется атрибут bgcolor="#XXXXXX", где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели следующим образом:



Какой цвет будет у страницы, заданной тэгом <body bgcolor="#00DDDD">?

- 1) белый
- 2) голубой
- 3) красный
- 4) черный

A16

Дан фрагмент электронной таблицы:

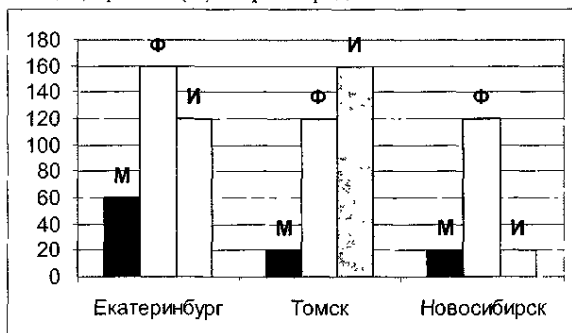
	A	B	C
1	2	=A\$1+C\$2	6
2	1	=C\$1+A\$2	3
3		=-1*C\$1	

После выполнения вычислений, значения в столбце B отсортировали по возрастанию. Какое значение будет храниться в ячейке B2?

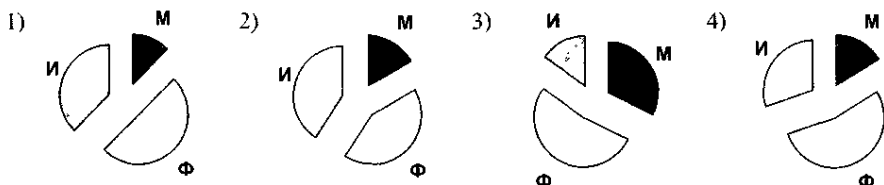
- 1) -6
- 2) 7
- 3) 5
- 4) 12

**A17**

На диаграмме показано количество призеров олимпиады по информатике (И), математике (М), физике (Ф) в трех городах России.



Какая из диаграмм правильно отражает соотношение призеров из всех городов по каждому предмету?



**A18**

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

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

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

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

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

Цикл

ПОКА < условие > команда

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

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

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

НАЧАЛО

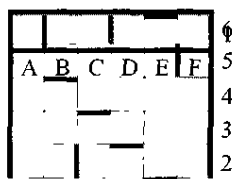
ПОКА < слева свободно > влево

ПОКА < снизу свободно > вниз

ПОКА < справа свободно > вправо

ПОКА < сверху свободно > вверх

КОНЕЦ



1) 1

2) 2

3) 3

4) 0

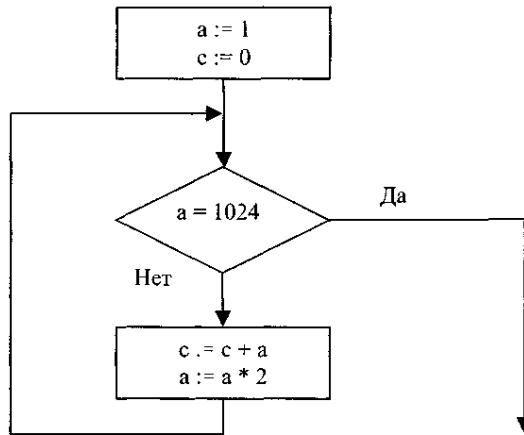
Ответом к заданиям этой части (В1 – В8) является набор символов, которые следует записать в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки. Каждый символ пишете в отдельной клеточке в соответствии с приведенными образцами.

**В1**

Одна ячейка памяти «троичной ЭВМ» (компьютера, основанного на использовании троичной системы счисления) может принимать одно из трех возможных состояний. Для хранения некоторой величины отвели 6 ячеек памяти. Сколько различных значений может принимать эта величина?

**В2**

Запишите значение переменной **c** после выполнения фрагмента алгоритма:



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

**В3**

Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 16 оканчивается на 1.

**В4**

$A, B, C$  – целые числа, для которых истинно высказывание  $(C > A) \wedge (C > B) \wedge (\neg(C - 1 > A) \vee \neg(C - 1 > B))$

Чему равно  $C$ , если  $A = 45, B = 18$ ?

**В5**

На экране есть два окна, в каждом из которых написано по числу  $U$  исполнителя Сумматор две команды, которым присвоены номера:

1. запиши сумму чисел в первое окно
2. запиши сумму чисел во второе окно

Выполняя первую из них, Сумматор складывает числа в окнах и заменяет этой суммой число в первом окне, а выполняя вторую, складывает числа и заменяет этой суммой число во втором окне. Запишите порядок команд в программе по-

лучения из пары чисел 1 и 2 пары чисел 13 и 4, содержащей не более 5 команд, указывая лишь номера команд.

(Например, программа **21211** это программа

**запиши сумму чисел во второе окно**

**запиши сумму чисел в первое окно**

**запиши сумму чисел во второе окно**

**запиши сумму чисел в первое окно**

**запиши сумму чисел в первое окно**

которая преобразует пару чисел 1 и 0 в пару чисел 8 и 3 )

**B6**

Классный руководитель пожаловался директору, что у него в классе появилась компания из 3-х учеников, один из которых всегда говорит правду, другой всегда лжет, а третий говорит через раз то ложь, то правду. Директор знает, что их зовут Коля, Саша и Миша, но не знает, кто из них правдив, а кто – нет. Однажды все трое прогуляли астрономию. Директор знает, что никогда раньше никто из них не прогуливал астрономию. Он вызвал всех троих в кабинет и поговорил с мальчиками. Коля сказал: "Я раньше никогда не прогуливал астрономию". Саша сказал: "Бывает, что я говорю неправду". Миша сказал: "Коля соврал Вам. Саша никогда не врет". Директор понял, кто из них кто. Расположите первые буквы имен мальчиков в порядке: "говорит всегда правду", "всегда лжет", "говорит правду через раз" (Пример: если бы имена мальчиков были Рома, Толя и Вася, ответ мог бы быть РТВ)

**B7**

Скорость передачи данных через ADSL-соединение равна 64000 бит/с. Через данное соединение передают файл размером 375 килобайт. Определите время передачи файла в секундах.

**B8**

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 сначала 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) БАА
- (3) СВААВАА
- (4) DCBAABAACBAABAА

*Латинский алфавит (для справки):*

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Запишите три символов подряд, стоящие в восьмой строке со 252-го по 254-е место (считая слева направо).

**B9**

Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес.

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

.64	3.13	3.133	40
А	Б	В	Г

**В10** В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

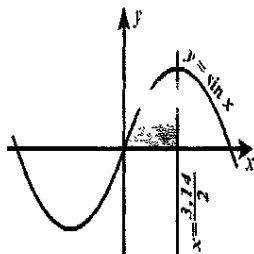
Для обозначения логической операции “ИЛИ” в запросе используется символ |, а для логической операции “И” – &.

1	принтеры   сканеры   продажа
2	продажа & принтеры
3	принтеры & сканеры & продажа
4	принтеры & сканеры & продажа & сервис

### Часть 3

*Для записи ответов к заданиям этой части (C1 – C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1 и т.д.), а затем полное решение. Ответы записывайте четко и разборчиво.*

**C1**



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

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre> var x, y real; begin   readln(x,y);   if y&lt;=sin(x) then     if x&lt;=3.14/2 then       if y&gt;=0 then         write('принадлежит')       else         write('не принадлежит')       end.     end.   end.</pre>	<pre> INPUT x, y IF y&lt;=SIN(x) THEN   IF x&lt;=3.14/2 THEN     IF y&gt;=0 THEN       PRINT "принадлежит"     ELSE       PRINT "не принадлежит"     ENDIF   ENDIF ENDIF END</pre>	<pre> void main(void) { float x,y;   scanf("%f%f",&amp;x,&amp;y);   if (y&lt;=sin(x))     if (x&lt;=3.14/2)       if (y&gt;=0)         printf("принадлежит");       else         printf("не принадлежит");     } }</pre>

Последовательно выполните следующее:

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

**C2**

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

**C3**

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

**C4**

На вход программе подается последовательность символов, среди которых могут быть и цифры, отличные от нуля. Ввод символов заканчивается точкой (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 составит из тех цифр, которые не встречаются во входных данных, минимальное число (ноль не используется). Каждая цифра при этом используется ровно один раз. Если во входных данных встречаются все цифры от 1 до 9, то следует вывести "0".

Например, пусть на вход подаются следующие символы  
1A734B39.

В данном случае программа должна вывести  
2568

## Вариант 2

**A1**

Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде Unicode, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 960 бит. Какова длина сообщения в символах?

- 1) 60                      2) 960                      3) 240                      4) 120

**A2**

В велокроссе участвуют 107 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 50 велосипедистов?

- 1) 50 бит                      2) 50 байт                      3) 350 бит                      4) 107 байт

**A3**

Дано  $a=3D_{16}$ ,  $b=77_8$ . Какое из чисел  $c$ , записанных в двоичной системе, отвечает условию  $a < c < b$ ?

- 1) 111101      2) 111110      3) 111111      4) 111010

**A4**

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

- 1)  $1010101_2$       2)  $125_8$       3)  $A3_{16}$       4)  $75_{16}$

**A5**

Переменные  $a$  и  $b$  описаны в программе как целочисленные. Определите значение переменной  $a$  после выполнения следующего фрагмента программы:

Бейсик	Паскаль	Алгоритмический
$a = 432$ $b = a \setminus 100$ $a = (a \text{ MOD } 100) * 10$ $a = a + b$ $\setminus$ и MOD – операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно	$a := 432;$ $b := a \text{ div } 100;$ $a := (a \text{ mod } 100) * 10;$ $a := a + b;$ $\{ \text{div и mod – операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно} \}$	$a = 432$ $b := \text{div}(a, 100)$ $a := \text{mod}(a, 100) * 10$ $a. = a + b$ $  \text{div и mod – функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}  $

- 1) 72      2) 44      3) 324      4) 752

**A6**

Дан фрагмент программы, обрабатывающей массив  $A$  из  $n$  элементов:

Бейсик	Паскаль	Алгоритмический
$j = 1$ FOR $i = 1$ TO $n$ IF $A(i) > A(j)$ THEN $j = i$ NEXT $i$ $s = A(j)$	$j := 1;$ for $i := 1$ to $n$ do begin if $A[i] > A[j]$ then $j := i$ end; $s := A[j]$	$j := 1$ <u>нц</u> для $i$ от 1 до $n$ <u>если</u> $A[i] > A[j]$ <u>то</u> $j := i$ <u>все</u> <u>кц</u> $s := A[j]$

Чему будет равно значение переменной  $s$  после выполнения данного алгоритма при любых значениях элементов массива  $A$ ?

- Максимальному элементу в массиве  $A$
- Индексу максимального элемента в массиве  $A$  (первому из них, если максимальных элементов несколько)
- Индексу максимального элемента в массиве  $A$  (последнему из них, если максимальных элементов несколько)
- Количеству элементов, равных максимальному в массиве  $A$

Для какого из указанных значений X истинно высказывание

$$((X>5) \rightarrow (X>7)) \wedge (\neg (X>4) \vee (X>5))$$

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

Укажите, какое логическое выражение равносильно выражению

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

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

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

Дан фрагмент таблицы истинности выражения F.

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

Какое выражение соответствует F?

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

Между четырьмя крупными аэропортами, обозначенными кодами FUJ, LAI, MPU и URP, ежедневно выполняются авиарейсы. Приведён фрагмент расписания перелётов между этими аэропортами.

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
LAI	FUJ	04:15	16:25
FUJ	LAI	07:20	08:45
URP	MPU	08:20	10:15
LAI	URP	08:30	10:45
FUJ	MPU	11:15	13:00
MPU	FUJ	13:15	15:00
MPU	URP	14:30	15:55
FUJ	URP	14:50	17:05
URP	FUJ	15:35	18:55
URP	LAI	21:40	23:10

Путешественник находится в аэропорту FUJ в полночь (0:00). Определите самое раннее время, когда он может оказаться в аэропорту URP.

- 1) 10:45                      2) 15:55                      3) 17:05                      4) 18:55



**A11**

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

Г	Д	К	О	Р
001	11	01	000	10

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

- 1) 011000100001100001
- 2) 011000100001100000
- 3) 001000100001100000
- 4) 001000100001100001

**A12**

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

Какая из перечисленных цепочек создана по этому правилу?

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

**A13**

Для групповых операций с файлами используются **маски имен файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «\*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «\*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске:

**b\*a\*c?.c?\***

- 1) bacc.cpp
- 2) bac.cpp
- 3) bacc c
- 4) blarc.cpp

**A14**

Сколько записей в нижеследующем фрагменте экзаменационной ведомости удовлетворяют условию

«(Пол='ж' И Физика<5) ИЛИ Алгебра=4»?

Номер	Экзаменуемый	Пол	Алгебра	Сочинение	Физика	История
1	Аксенов	м	5	4	5	3
2	Андреев	м	3	5	4	5
3	Васильева	ж	3	5	4	5
4	Кондратьев	м	4	5	4	5
5	Лисова	ж	4	3	3	4
6	Прокопьев	м	3	2	4	3

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

**A15**

Для кодирования цвета фона страницы Интернет используется атрибут bgcolor="#XXXXXX", где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели следующим образом.



Какой цвет будет у страницы, заданной тэгом <body bgcolor="#818181">?

- 1) белый                      2) черный                      3) голубой                      4) серый

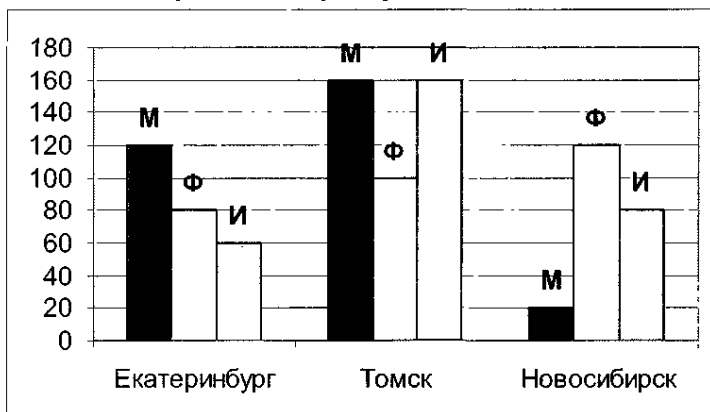
**A16**

В электронной таблице значение формулы =СРЗНАЧ(С1:С3) равно 4. Значение формулы =СРЗНАЧ(С4:С5) равно 6. Чему будет равно значение формулы =СУММ(С1:С5)?

- 1) 5                              2) 24                              3) 25                              4) 10

**A17**

На диаграмме показано количество призов олимпиады по информатике (И), математике (М), физике (Ф) в трех городах России.



Какая из диаграмм правильно отражает соотношение призов из всех городов по каждому предмету?

- 1) 2) 3) 4)

**A18**

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

<b>вверх</b>	<b>вниз</b>	<b>влево</b>	<b>вправо</b>
--------------	-------------	--------------	---------------

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:

<b>сверху свободно</b>	<b>снизу свободно</b>	<b>слева свободно</b>	<b>справа свободно</b>
------------------------	-----------------------	-----------------------	------------------------

Цикл

ПОКА < условие > команда

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

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

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

НАЧАЛО

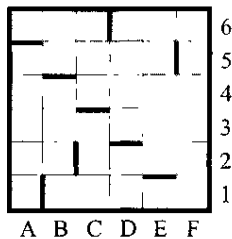
ПОКА < снизу свободно > вниз

ПОКА < справа свободно > вправо

ПОКА < сверху свободно > вверх

ПОКА < слева свободно > влево

КОНЕЦ



1) 1

2) 2

3) 3

4) 0

Часть 2

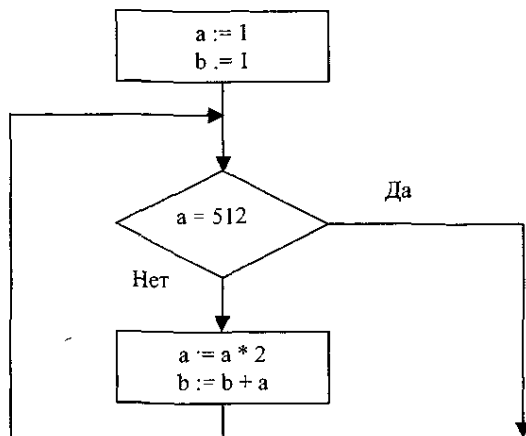
Ответом к заданиям этой части (B1 – B8) является набор символов, которые следует записать в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки. Каждый символ пишете в отдельной клеточке в соответствии с приведенными образцами.

**B1**

Для передачи сигналов на флоте используются специальные сигнальные флаги, вывешиваемые в одну линию (последовательность важна). Какое количество различных сигналов может передать корабль при помощи двух сигнальных флагов, если на корабле имеются флаги шести различных видов (флагов каждого вида неограниченное количество)?

**B2**

Запишите значение переменной **b** после выполнения фрагмента алгоритма:



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

**B3**

Укажите через запятую в порядке возрастания все числа, не превосходящие 11, запись которых в пятеричной системе счисления начинается на 2.

**B4**

Укажите значения переменных  $K, L, M, N$ , при которых логическое выражение  $(K \rightarrow M) \wedge (\neg L \vee M) \wedge N \wedge \neg M$  истинно.

Ответ запишите в виде строки из четырех символов: значений переменных  $K, L, M$  и  $N$  (в указанном порядке). Так, например, строка 1111 соответствует тому, что  $K=1, L=1, M=1, N=1$ .

**B5**

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 3
2. умножь на 4

Выполняя первую из них, Калькулятор прибавляет к числу на экране 3, а выполняя вторую, умножает его на 4. Запишите порядок команд в программе получения из числа 3 числа 57, содержащей не более 6 команд, указывая лишь номера команд

(Например, программа 21211 это программа

умножь на 4

прибавь 3

умножь на 4

прибавь 3

прибавь 3

которая преобразует число 2 в 50.)

**B6**

Классный руководитель пожаловался директору, что у него в классе появилась компания из 3-х учеников, один из которых всегда говорит правду, другой всегда лжет, а третий говорит через раз то ложь, то правду. Директор знает, что их зовут Коля, Саша и Миша, но не знает, кто из них правдив, а кто – нет. Однажды все трое опоздали на урок. Он вызвал всех троих в кабинет и поговорил с мальчиками. Миша сказал: "Саша никогда не врёт. А вот от Коли, наоборот, никогда не услышишь правды". Саша сказал: "Миша сказал правду про меня". Директор понял, кто из них кто. Расположите первые буквы имен мальчиков в порядке "говорит всегда правду", "всегда лжет", "говорит правду через раз" (Пример: если бы имена мальчиков были Рома, Толя и Вася, ответ мог бы быть. РТВ)

**B7**

Скорость передачи данных через ADSL-соединение равна 64000 бит/с. Через данное соединение передают файл размером 625 килобайт. Определите время передачи файла в секундах

**B8**

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 сначала 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) БАА
- (3) СВААВАА
- (4) DCBAABAACBAABAА

*Латинский алфавит (для справки):*

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Запишите семь символов подряд, стоящие в восьмой строке со 128-го по 134-е место (считая слева направо).

**B9**

Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес.

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

1.14	3.110.	30	20
А	Б	В	Г

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

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

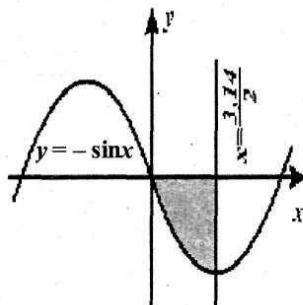
1	пловцы   гимнасты
2	пловцы & гимнасты & олимпиада
3	пловцы
4	пловцы & гимнасты

**Не забудьте перенести все ответы в бланк ответов № 1.**

### Часть 3

Для записи ответов к заданиям этой части (C1 – C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1 и т.д.), а затем полное решение. Ответы записывайте четко и разборчиво.

**C1** Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости ( $x, y$  – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно.



Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var x,y: real; begin   readln(x,y);   if x&lt;=3.14/2 then     if y&lt;=0 then       if y&gt;= - sin(x) then         write('принадлежит')       else         write('не принадлежит')     end.   end.</pre>	<pre>INPUT x, y IF x&lt;=3.14/2 THEN   IF y&lt;=0 THEN     IF y&gt;= - SIN(x) THEN       PRINT "принадлежит"     ELSE       PRINT "не принадлежит"     ENDIF   ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y;   scanf("%f%f",&amp;x,&amp;y);   if (x&lt;=3.14/2)     if (y&lt;=0)       if (y&gt;= - sin(x))         printf("принадлежит");       else         printf("не принадлежит");     } }</pre>

Последовательно выполните следующее:

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

**C2**

Опишите на русском языке или одном из языков программирования алгоритм подсчета максимальной длины последовательности непрерывно возрастающих значений элементов в заданном целочисленном массиве из 30 элементов.

**C3**

Два игрока играют в следующую игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. В начале игры фишка находится в точке с координатами (2,3). Ход состоит в том, что игрок перемещает фишку из точки с координатами (x,y) в одну из трех точек: или в точку с координатами (2x,y), или в точку с координатами (x,y+3), или в точку с координатами (x,y+4). Выигрывает игрок, после хода которого расстояние от фишки до точки с координатами (0,0) больше 14 единиц. Кто выигрывает при безошибочной игре обоих игроков – игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

**C4**

На вход программе подаются сведения о ячейках автоматической камеры хранения багажа. В первой строке задана текущая дата: через точку два целых числа, соответствующие дню (от 01 до 31 – ровно 2 символа) и месяцу (от 01 до 12 – ровно 2 символа). Во второй строке сообщается количество занятых ячеек N, которое не меньше 3, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат: <номер ячейки> <дата сдачи багажа>, где <номер ячейки> – четырехзначное число, <дата сдачи багажа> – через точку два целых числа, соответствующие дню (от 01 до 31 – ровно 2 символа) и месяцу (от 01 до 12 – ровно 2 символа). Номер ячейки и дата сдачи багажа разделены одним пробелом. Сведения отсортированы в порядке номеров ячеек.

Время хранения багажа – трое суток. Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая выведет номера ячеек, в которых багаж хранится заведомо больше трех суток – то есть разница между датой сдачи багажа и текущей датой составляет 4 и более дней. Номера ячеек следует выводить в хронологическом порядке сдачи багажа.

Багаж мог сдаваться только в текущем или предыдущем месяце текущего календарного года (если текущий месяц январь, то данные о сдаче багажа в декабре прошлого года отсутствуют). Количество дней в каждом из месяцев текущего года следующее: январь – 31, февраль – 28, март – 31, апрель – 30, май – 31, июнь – 30, июль – 31, август – 31, сентябрь – 30, октябрь – 31, ноябрь – 30, декабрь – 31. Все входные данные корректны.

Пример входных данных:

04.06

3

1000 01.06

1001 31.05

2007 21.05

Результат работы программы для этого примера

2007

1001

## ОТВЕТЫ И РЕШЕНИЯ

### Ответы к тренировочным заданиям

1.1 2 1.2 2 1.3 2 1.4 1 1.5 2 1.6 1 1.7 4 1.8 3 1.9 3 1.10 3 1.11 8 1.12 2 1.13 1 1.14 4 1.15 3  
 1.16 1 1.17 3 1.18 3 1.19 1 1.20 4 1.21 3 1.22 2 1.23 400 1.24 256 1.25 2400 1.26 50 1.27 128  
 1.28 4 1.29 2 1.30 2 1.31 3 1.32 4 1.33 2 1.34 2 1.35 3 1.36 2 1.37 11.38 4 1.39 3 1.40 4  
 1.41 3,5,15 1.42 5, 13, 21 1.43 3, 7,21 1.44 4 1.45 4 1.46 2 1.47 2 1.48 1 1.49 2 1.50 2  
 1.51 Назад 5 1.52 12211 1.53 121221 1.54 2212 1.55 4 1.56 170 1.57 1 1.58 4 1.59 4 1.60 2  
 1.61 414 1.62 142 1.63 3 1.64 170

1.65 Выигрывает первый игрок (при этом его первый ход должен быть 2,3,4 --> 4,5,6).

1.66 Выигрывает первый игрок, своим первым ходом он должен увеличить в 3 раза количество камней в первой куче.

1.67 Выигрывает второй игрок (при этом его первый ход должен быть 6,5)

1.68 Выигрывает первый игрок Своим первым ходом он должен удвоить количество камней во второй куче.

1.69 Выигрывает второй игрок После его первого хода координата x должна быть равна 8

1.70 3 1.71 2 1.72 2 1.73 2 1.74 3 1.75 2550 1.76 3 1.77 2 1.78 1 1.79 4 1.80 2 1.81 2 1.82 2  
 1.83 3 1.84 4 1.85 4 1.86 3 1.87 10 1.88 4 1.89 3990 1.90 3

### 1.91

**Содержание верного ответа и указания по оцениванию**

**(допускаются иные формулировки ответа, не искажающие его смысла)**

Введем целочисленную переменную Max, в которую будем заносить максимальный из просмотренных элементов массива, и целочисленную переменную Num, в которой будем подсчитывать число повторений максимального элемента в просмотренной части массива. В цикле до конца массива, сравниваем очередной элемент массива с текущим значением переменной Max, если он больше, то заносим его значение в Max и устанавливаем счетчик Num в 1, если они равны, то увеличиваем счетчик Num на 1. По окончании цикла переменная Num содержит число повторений максимального элемента массива.

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

На языке Паскаль

```
const N=30;
var a:array[1..N] of integer;
    Max, Num, i: integer;
begin
    Max:=a[1];
    Num:=1;
    for i:=2 to N do
        begin
            if a[i]>Max then
                begin
                    Max:=a[i];
                    Num:=1;
                end
            else
                if a[i]=Max then
                    Num:=Num+1;
                end
        end;
    writeln(Num);
end
```

На языке Бейсик

```
N=30
DIM i, Max, Num, a(N) AS INTEGER
Max=a(1)
Num=1
LOOP
FOR i = 2 TO N
IF a(i)>Max THEN
    Max=a(i)
    Num=1
ELSE
    IF a(i)=Max THEN
        Num=Num+1
    ENDIF
ENDIF
NEXT i
PRINT Num
END
```



**указания по оцениванию  
логики ответа, не искажающие его смысла)**

и MaxProd, в которую будем заносить максимальное произведение элементов в просмотренной части массива, и переменную MaxNum, равную произведению первого элемента в этой паре. Первоначально в эти переменные заносим первый элемент и номер 1 соответственно. В цикле до конца массива вычисляем произведение элементов очередной пары уже найденный максимум и заносим в MaxProd новое произведение, а в переменную MaxNum – максимум из текущего значения и произведения. По окончании цикла выводим значение переменной MaxNum. Это решение является частью активной программы (на основе алгоритма, использующего однократ-

	На языке Бейсик	
	<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)&gt;MaxProd THEN     MaxNum=i     MaxProd=a(i)*a(i+1) ENDIF NEXT i PRINT MaxNum END </pre>	

**указания по оцениванию  
форматирования ответа, не искажающие его смысла)**

и переменную SumProd, в которую будем заносить сумму произведений элементов в просмотренной части массива, и присвоим ей начальное значение. В цикле считаем произведение элементов очередной пары и прибавляем его к текущему значению SumProd. По окончании цикла переменная SumProd содержит сумму произведений элементов массива. Это решение является частью активной программы (на основе алгоритма, использующего однократ-

	На языке Бейсик	
<pre> i-1]*a[2*i]; </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>	

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

Введем целочисленные переменные Max, Min и Sum, в которые будем заносить соответственно значения максимального и минимального элемента в просмотренной части массива, а также накапливать сумму значений элементов. Присвоим им в качестве начального значение первого элемента массива. Также заводим две переменные SM и SA типа real для хранения средних значений. В цикле от второго элемента до конца массива прибавляем элемент к сумме, сравниваем его с Max, если он больше, заносим его значение в переменную Max. В противном случае сравниваем его с Min, если он меньше, заносим его значение в переменную Min. По окончании цикла вычисляем среднее арифметическое Max и Min, заносим его в переменную SM. В переменную SA заносим частное от деления суммы элементов на количество элементов в массиве. Выводим разность SM - SA.

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; Max, Min, Sum, I: integer; SM, SA: real; begin   Max := a[1];   Min := a[1];   Sum := a[1];   for I := 2 to N do     begin       Sum := Sum + a[I];       if a[I] &gt; Max then         Max := a[I];       else         if a[I] &lt; Min then           Min := a[I];         end       SM := (Max + Min)/2;       SA := Sum/N;       writeln (SM - SA);     end.</pre>	<pre>N=30 DIM I, Max, Min, Sum A(N) AS INTEGER DIM SA, SM AS REAL Max = A(1) Min = A(1) Sum = A(1) FOR I = 2 TO N   Sum = Sum + A(I)   IF A(I) &gt; Max THEN     Max = A(I)   ELSE     IF A(I) &lt; Min THEN       Min = A(I)     ENDIF   ENDIF NEXT I SM = (Max + Min)/2 SA = Sum/N PRINT SM-SA END</pre>

1.95 3 1.96 3 1.97 4 1.98 1 1.99 2 1.100 4 1.101 4 1.102 1 1.103 1 1.104 3 1.105 X V Y  
1.106 X A Y 1.107 3 1.108 4 1.109 4 1.110 4 1.111 1100 1.112 4 1.113 19 1.114 17  
1.115 44 1.116 B 1.117 M 1.118 Коля 1.119 СКМ  
1.120 3 1.121 4 1.122 2 1.123 3 1.124 3

2.1 2 2.2 4 2.3 P:\doc\txt\txt.doc 2.4 1 2.5 4 2.6 2 2.7 1 2.8 4 2.9 4 2.10 1

2.11 1 2.12 512 2.13 256 2.14 1 2.15 2 2.16 2 2.17 =A2+2\*\$D1 2.18 1 2.19 4 2.20 5  
2.21 4 2.22 2 2.23 4 2.24 1 2.25 1 2.26 1 2.27 4 2.28 3 2.29 3 2.30 4 2.31 1 2.32 4 2.33 3  
2.34 home.net 2.35 АББДЖЕГ

2.36 ДБГЖВАЕ 2.37 БГЖДВЕА 2.38 БГАВ 2.39 1 2.40 (большая & белая & акула) | кар-харолон

## 2.45

Элементы ответа:

- 1) Поле  $i=5, j=4$  будет выдано приведенной программой (т.к.  $i+j=9$ )
- 2)  $(i=1, j=8), (i=2, j=8), (i=5, j=5)$  – эти поля удовлетворяют постановке задачи (ответ очевиден, если нарисовать шахматную доску  $8 \times 8$  и королей в двух нижних углах)
- 3) Возможный (самый короткий) способ доработки вместо `if (i=9-j) OR (i=j)` проверять условие `if (i>=9-j) AND (i<=j)`

## 2.46

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).

## 2.47

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

Элементы ответа:

- 1) Пример  $x=3, y=0,5$  (Любая пара  $(x,y)$ , для которой выполняется:  $y>1$  или  $x<0$  или  $(y>=\sin x$  и  $x>\pi/2$  и  $y<=1)$ )

2) Возможная доработка (Паскаль):

`if (y<=1) and (x>=0) and (y>=sin(x)) and (x<=3,14/2) then`

`write('принадлежит')`

`else`

`write('не принадлежит')`

(могут быть и другие способы доработки).

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Программа читает текст из файла один раз, сразу подсчитывая в массиве, хранящем 12 вещественных чисел, сумму температур в каждом из месяцев, одновременно суммируя все температуры в году. Затем с использованием этого массива ищется минимальное отклонение среднемесячной температуры от среднегодовой. За дополнительный просмотр этого массива распечатывается информация об искомым месяцах. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, месяц с минимальным отклонением единственен).</p>	
Указания по оцениванию	Баллы
<p>Программа работает верно, т.е. определяет все месяцы, в которых среднемесячная температура минимально отклоняется от среднегодовой, не содержит вложенных циклов, в тексте программы не анализируется каждый месяц в отдельности. Допускается наличие в тексте программы одной пунктуационной ошибки. Пример правильной и эффективной программы.</p> <pre> Const d:array[1..12] of integer = (31,29,31,30,31,30,31,31,30,31,30,31); var tm:array[1..12] of real;     m:1..12;     data:string[5],     min,ty,t:real;     i:integer; begin   for i:=1 to 12 do     tm[i]:=0;   ty:=0;   for i:=1 to 366 do     begin       readln(data,t);       m:=(ord(data[4])-ord('0'))*10         +ord(data[5])-ord('0');       tm[m]:=tm[m]+t;       ty:=ty+t;     end;   for i:=1 to 12 do     tm[i]:=tm[i]/d[i];   ty:=ty/366;   min:=100;   for i:=1 to 12 do     if abs(tm[i]-ty)&lt;min then       min:=abs(tm[i]-ty);   writeln(ty:0:2);   for i:=1 to 12 do     if abs(abs(tm[i]-ty)-min)&lt;0.0001 then       writeln(i,' ',tm[i]:0:2,' ',abs(tm[i]-ty):0.2);   readln end.</pre>	4

Программа работает верно, но содержит вложенные циклы (от 1 до 12 и от 1 до 366). 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.	3
Программа работает в целом верно, но выводит значение только одного месяца с минимальным отклонением температуры. Допускается наличие от одной до пяти синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования, не описана или неверно описана переменная.	2
Программа обрабатывает каждый месяц явным образом (12 операторов IF или оператор CASE, содержащий 12 вариантов), и, возможно, неверно работает при некоторых входных данных (например, содержит ошибку в алгоритме поиска минимума, в результате которой не всегда находятся нужные элементы или отклонение берется не по абсолютной величине). Допускается наличие от одной до семи синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования, не описана или неверно описана переменная.	1
Задание не выполнено или выполнено неверно	0
<i>Максимальный балл</i>	4

## 2.49

см. "методические рекомендации..." п.2.8

**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 невозрастанию 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. 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
```

## 2.51

**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 часа. Время при считывании удобно перевести в минуты и в этом же виде хранить и сравнивать. Затем полученный массив времен сортируется по неубыванию любым алгоритмом сортировки, параллельно переставляются и элементы массива с фамилиями (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,cl:pp;
    c,cl:char;
    i,j,N,time1:integer;
begin
    read(c,cl), {считаны часы текущего времени}
    time1:=60*((ord(c)-ord('0'))*10+ord(cl)-ord('0'));
    readln(c,c,cl); {пропущено двоеточие, и считаны минуты}
    time1:=time1+(ord(c)-ord('0'))*10+ord(cl)-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,cl), {считаны часы первого времени}
        p[j].time:=60*((ord(c)-ord('0'))*10+ord(cl)-ord('0'));
        readln(c,c,cl); {пропущено двоеточие, и считаны минуты}
        p[j].time:=p[j].time+(ord(c)-ord('0'))*10+ord(cl)-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;
        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 + 1) = time2: m(j + 1) = nm
END IF
NEXT j
NEXT i
FOR i = 1 TO k
PRINT m(i)
NEXT i
END
```

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

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

**Указания по оцениванию**

Программа работает верно, т.е. корректно выделяет из входных данных номера школ, не содержит вложенных циклов, в тексте программы не анализируется каждая школа в отдельности, все считанные номера не запоминаются в массиве. Допускается наличие в тексте программы одной синтаксической ошибки. Пример правильной и эффективной программы:

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

# Ответы к тренировочным вариантам-2009

## Ответы к варианту 1

### ЧАСТЬ 1

№ задания	Ответ	№ задания	Ответ
A1	1	A10	1
A2	4	A11	1
A3	1	A12	2
A4	2	A13	1
A5	3	A14	4
A6	2	A15	2
A7	3	A16	3
A8	4	A17	1
A9	1	A18	2

### ЧАСТЬ 2

№	Ответ
B1	729
B2	1023
B3	3,5,15
B4	46
B5	22111
B6	KMC
B7	48
B8	ABA
B9	ГБВА
B10	4321

### ЧАСТЬ 3

C1

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

Элементы ответа

1) Пример.  $x = -4$ ,  $y = 0.5$  (Любая пара  $(x, y)$ , для которой выполняется  $y > \sin x$  или  $x > 3.14/2$  или  $(y < \sin x$  и  $x < 0$  и  $y > 0)$ )

2) Возможная доработка (Паскаль)

```
if (y <= sin(x)) and (x <= 3.14/2) and (y >= 0) and (x >= 0) then
  write('принадлежит')
else
  write('не принадлежит')
```

(могут быть и другие способы доработки)

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить <b>три</b> действия: указать пример входных данных, при которых программа работает неверно и исправить две ошибки.</p> <p>1 Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE).</p> <p>2 Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых <math>(y \leq \sin x)</math> и <math>(y = 0)</math> и <math>(x &lt; 0)</math>.</p>	
<p>Правильно выполнены оба пункта задания. Исправлены обе ошибки.</p> <p>В работе (во фрагментах программ) допускается не более одной синтаксической ошибки</p>	3
<p>Правильно выполнены 2 пункта задания из трех (исправлены обе ошибки, но не указан/неправильно указан пример требуемых входных данных, либо правильно указан пример входных данных, программа правильно работает при большем числе случаев, чем исходная, но не при всех).</p> <p>Например, выдает "принадлежит" для точек, у которых <math>(y \leq \sin x)</math> и <math>(y = 0)</math> и <math>(x &lt; 0)</math>.</p> <p>Допускается, например, такое решение:</p> <pre> if y&lt;=sin(x) then   if x&lt;=3.14/2 then     if y&gt;=0 then       write('принадлежит')     else       write('не принадлежит')     else       write('не принадлежит')   else     write('не принадлежит') else   write('не принадлежит') </pre> <p>При этом в данной работе допускается не более двух синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)</p>	2
<p>Правильно выполнен только один пункт задания.</p> <p>То есть, только приведен пример входных данных, либо он не приведен, но имеется программа, корректно работающая при большем количестве входных данных, чем исходная.</p> <p>При этом, если приведена программа, то в ней допускается не более трех синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)</p>	1
<p>Все пункты задания выполнены неверно (пример входных данных не указан или указан неверно, программа не приведена, либо приведенная программа корректно работает в не большем количестве случаев, чем исходная)</p>	0
Максимальный балл	3

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

Заводим новый целочисленный массив той же длины. В цикле от первого элемента до последнего сравниваем элементы исходного массива с нулем и отрицательным элементом меняем знак. Записываем значения в элементы второго массива с тем же номером. Печатать значения массива не обязательно.

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, b:array[1..N] of integer; i: integer; begin   for i:=1 to N do     if a[i] &lt; 0     then b[i]:= - a[i]     else b[i] = a[i];   end.</pre>	<pre>N=30 DIM I, A(N), B(N) AS INTEGER FOR I = 1 TO N   IF A(I) &lt; 0 THEN     B(I) = - A(I)   ELSE     B(I) = A(I)   ENDIF NEXT I END</pre>

Указания по оцениванию	Баллы
Предложен правильный алгоритм, выдающий верное значение. Возможно использование числа 30 вместо константы. Возможно использование операции "больше" (Так как $-0=0$ ) Возможно использование двух индексных переменных 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 объявлен массив результатов 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.	0
<i>Максимальный балл</i>	2

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

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

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

Таблица содержит *все возможные* варианты ходов второго игрока. Из неё 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 доказательства ее правильности.	2
При наличии в представленном решении одного из пунктов: 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 рассмотрены несколько (больше одного, но не все) вариантов ответов второго игрока.	1
Задание не выполнено или 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
<i>Максимальный балл</i>	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, помечая в массиве, состоящем из 9 элементов, какие цифры встретились во входных данных. Сами цифры при этом не запоминаются. За дополнительный проход этого массива печатаются те цифры, которые оказались не помеченными, в противном случае выводится 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, или когда в результате получаются небольшие числа).

**Пример правильной и эффективной программы на языке Паскаль:**

```
var a: array[1..9] of boolean;
    c: char;
    i, k: integer;
begin
  for i:= 1 to 9 do
    a[i]:=false;
  read(c);
  while c<>'.' do
    begin
      k:=ord(c)-ord('0');
      if (k>0) and (k<10) then a[k]:=true;
      read(c);
    end;
  k:=0;
  for i:=1 to 9 do
    if not a[i] then
      begin
        k:=k+1;
        write(i)
      end;
    if k=0 then write(0);
  writeln
end.
```

**Пример правильной и эффективной программы на языке Бейсик:**

```
DIM i, j, k, a(9) AS INTEGER
FOR i = 1 TO 9
  a(i) = 0
NEXT
INPUT c$
DO WHILE NOT (c$ = ".")
  j = ASC(c$)
  j = j - ASC("0") + 1
  IF (j>0) AND (j<10) THEN a(j) = a(j) + 1
  INPUT c$
```

<pre> LOOP k = 0 FOR i = 1 TO 9   IF a(i) = 0 THEN     k = k + 1     PRINT CHR\$(i + 48);   END IF NEXT IF k = 0 THEN PRINT 0 END </pre>	
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 не анализируется каждая цифра в отдельности. Допускается наличие в тексте программы одной пунктуационной ошибки.	4
Программа работает верно, но входные данные запоминаются в массиве символов или строке, или входные данные считываются несколько раз. Возможно, каждая цифра обрабатывается явным образом (9 операторов IF, в том числе с использованием многоточия при записи программы, или оператор CASE, содержащий 9 – 10 вариантов). Возможно, после сохранения входных данных для каждой цифры они просматриваются заново и анализируется наличие соответствующей цифры, или из отобранных цифр формируется число вместо печати этих цифр в порядке возрастания. В программе присутствуют вложенные циклы (один по входным данным, второй – по цифрам, он может быть заменен оператором CASE или 9 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, недопустимая для соответствующего типа данных.	3
Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержится 1 – 2 ошибки (выход за границу массива, перевод символов в числа, из отобранных цифр формируется число, которое не помещается в используемый тип данных, например, 16-битное целое и т.п.). 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, недопустимая для соответствующего типа данных.	2
Программа, возможно, неверно работает при некоторых входных данных. Возможно, программа не определяет или неверно определяет, что все цифры во входных данных встречались, 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, недопустимая для соответствующего типа данных.	1
Задание не выполнено или выполнено неверно	0
Максимальный балл	4



## Ответы к варианту 2

### ЧАСТЬ 1

№ задания	Ответ	№ задания	Ответ
A1	4	A10	2
A2	3	A11	4
A3	2	A12	1
A4	4	A13	1
A5	3	A14	3
A6	1	A15	4
A7	4	A16	2
A8	1	A17	1
A9	2	A18	2

### ЧАСТЬ 2

№	Ответ
B1	36
B2	1023
B3	2,10,11
B4	1
B5	22111
B6	KMC
B7	80
B8	AGFEDCB
B9	ГАБВ
B10	2431

### ЧАСТЬ 3

C1

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

Элементы ответа.

1) Пример:  $x = -4$ ,  $y = -0,5$  (Любая пара  $(x, y)$ , для которой выполняется.  $x > 3,14/2$  или  $y > 0$  или  $(y \geq -\sin x$  и  $x < 0$  и  $y < 0)$ )

2) Возможная доработка (Паскаль):

```
if (y >= -sin(x)) and (x <= 3 14/2) and (y <= 0) and (x >= 0) then
  write('принадлежит')
else
  write('не принадлежит')
```

(могут быть и другие способы доработки).

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить <b>три</b> действия: указать пример входных данных, при которых программа работает неверно, и исправить две ошибки:</p> <p>1. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE)</p> <p>2. Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых <math>(y &gt;= -\sin x)</math> и <math>(y &lt;= 0)</math> и <math>(x &lt; 0)</math></p>	
<p>Правильно выполнены оба пункта задания. Исправлены обе ошибки.</p> <p>В работе (во фрагментах программ) допускается не более одной синтаксической ошибки</p>	3
<p>Правильно выполнены 2 пункта задания из трех (исправлены обе ошибки, но не указан/неправильно указан пример требуемых входных данных, либо правильно указан пример входных данных, программа правильно работает при большем числе случаев, чем исходная, но не при всех).</p> <p>Например, выдает "принадлежит" для точек, у которых <math>(y &gt;= -\sin x)</math> и <math>(y &lt;= 0)</math> и <math>(x &lt; 0)</math>.</p> <p>Допускается, например, такое решение:</p> <pre>if x&lt;=3.14/2 then   if y&lt;=0 then     if y&gt;= -sin(x) then       write('принадлежит')     else       write('не принадлежит')</pre> <p>else   write('не принадлежит')</p> <p>При этом в сданной работе допускается не более двух синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)</p>	2
<p>Правильно выполнен только один пункт задания</p> <p>То есть, только приведен пример входных данных, либо он не приведен, но имеется программа, корректно работающая при большем количестве входных данных, чем исходная.</p> <p>При этом, если приведена программа, то в ней допускается не более трех синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)</p>	1
<p>Все пункты задания выполнены неверно (пример входных данных не указан или указан неверно, программа не приведена, либо приведенная программа корректно работает в не большем количестве случаев, чем исходная)</p>	0
<p>Максимальный балл</p>	3

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

Введем целочисленную переменную LenSQ с первоначальным значением, равным единице, которую будем увеличивать на единицу при возрастании значений элементов. Введем также переменную MaxLenSQ для записи результата вычислений. В цикле от второго элемента до конца массива сравниваем текущий элемент с предыдущим. В случае, если последующий элемент больше предыдущего, увеличиваем переменную LenSQ на единицу. В противном случае, во-первых, сравниваем значения переменных MaxLenSQ и LenSQ. В случае если значение переменной LenSQ больше, заносим его в переменную MaxLenSQ. Во-вторых, устанавливаем значение переменной LenSQ равным единице. Переходим к следующему элементу. По окончании цикла еще раз сравниваем MaxLenSQ и LenSQ. Затем выводим значение переменной MaxLenSQ.

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; LenSQ, MaxLenSQ, i: integer; begin   LenSQ:=1;   MaxLenSQ:=1;   for i:=2 to N do   begin     if a[i] &gt; a[i-1]     then LenSQ = LenSQ + 1     else begin       if LenSQ &gt; MaxLenSQ       then MaxLenSQ:=LenSQ;       LenSQ:=1;     end   end;   if LenSQ&gt;MaxLenSQ   then MaxLenSQ:=LenSQ;   writeln(MaxLenSQ); end </pre>	<pre> N=30 DIM I, LenSQ, MaxLenSQ, A(N) AS INTEGER LenSQ = 1 MaxLenSQ = 1 FOR I = 2 TO N   IF A(I) &gt; A(I-1) THEN     LenSQ = LenSQ + 1   ELSE     IF LenSQ &gt; MaxLenSQ THEN       MaxLenSQ = LenSQ     END IF   END IF   LenSQ = 1 NEXT I IF LenSQ&gt;MaxLenSQ THEN   MaxLenSQ = LenSQ PRINT MaxLenSQ END </pre>

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение, в том числе алгоритм, предусматривающий многократный проход по массиву.</p> <p>Возможно использование числа 30 вместо константы</p> <p>Может не быть сравнения MaxLenSQ и LenSQ после завершения цикла.</p> <p>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> <ol style="list-style-type: none"> <li>1) Не задано или неверно задано начальное значение LenSQ</li> <li>2) Не задано начальное значение MaxLenSQ</li> <li>3) Значение переменной LenSQ не уменьшается до единицы при обрыве последовательности</li> <li>4) Не указано или неверно указано условие завершения цикла</li> <li>5) Индексная переменная в цикле не меняется</li> <li>6) Неверно расставлены операторные скобки</li> <li>7) Программа не выводит или не возвращает результат</li> </ol>	1
Ошибок, перечисленных выше, больше двух, или алгоритм сформулирован неверно.	0
<i>Максимальный балл</i>	2

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

Выигрывает первый игрок, своим первым ходом он должен поставить фишку в точке с координатами (2,6). Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке координаты фишки на каждом этапе игры

1 ход	2 ход	3 ход	4 ход	5 ход
Позиция после первого хода	II-й игрок (все варианты хода)	I-й игрок (выигрышный ход)	II-й игрок (все варианты хода)	I-й игрок (один из вариантов)
<u>2,6</u>	4,6	<u>4,9</u>	8,9	<u>16,9</u>
			4,12	<u>8,12</u>
			4,13	<u>8,13</u>
	2,9	<u>4,9</u>	Те же варианты 4-5 ходов	
	2,10	<u>2,14</u>	Первый игрок выигрывает ответным ходом	

Таблица содержит *все возможные* варианты ходов второго игрока. Из неё 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 доказательства ее правильности	2
При наличии в представленном решении одного из пунктов: 1. Правильно указаны все варианты хода первого игрока и возможные ответы второго игрока (в том числе, и все выигрышные), но неверно определены дальнейшие действия и неправильно указан победитель. 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.	1
Задание не выполнено или 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
<i>Максимальный балл</i>	3

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

Программа верно читает входные данные, сразу запоминая в массиве номера ячеек и времена сдачи багажа только для тех ячеек, для которых допустимое время хранения истекло. Затем полученный массив дат сортируется по неубыванию любым алгоритмом сортировки, параллельно переставляются и элементы массива с номерами ячеек (возможно использование одного массива записей, состоящих из двух-трех полей) Печатаются элементы массива номеров ячеек в полученном в результате сортировки порядке. Примеры программ на языках Паскаль и Бейсик демонстрируют два возможных подхода к считыванию данных и их хранению.

**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);
type pp=record
  num, day,mon: integer;
end;
function diff(a,b:pp) integer;
{определяет разницу между двумя датами}
begin
  if a.mon=b.mon then diff:=b.day-a.day else
    diff:=b.day+(d[a.mon]-a.day)
end;
var
  p:array[1..1000] of pp;
  q:pp;
  c,c1:char;
  i,j,N,day1,mon1: integer;
begin
  read(c,c1); {считан день текущей даты}
  q.day := (ord(c)-ord('0'))*10+ord(c1)-ord('0');
  readln(c,q.mon); {пропущена точка и считан месяц}
  readln(N);
  j:=1;
  for i=1 to N do
  begin
    read(p[j].num);
    read(c,c,c1); {считан день даты}
    p[j].day:=(ord(c)-ord('0'))*10+ ord(c1)-ord('0');
    readln(c,p[j].mon); {пропущена точка, считан месяц}
    if diff(p[j],q)>3 then {данные занесены в массив}
```

```

j:=j+1;
end;
N:=j-1;
for i:=1 to N-1 do {сортируем данные}
  for j:=1 to N-i do
    if (p[j].mon>p[j+1].mon) or
      (p[j].mon=p[j+1].mon)and(p[j].day>p[j+1].day)then
      begin
        q:=p[j];
        p[j]:=p[j+1];
        p[j+1]:=q;
      end;
  end;
for i:=1 to N do
  writeln(p[i].num)
end.

```

Пример правильной программы на языке Бейсик:

```

DECLARE FUNCTION diff (d1, m1, d2, m2)
DIM SHARED d(12) AS INTEGER
DATA 31,28,31,30,31,30,31,31,30,31,30,31
FOR i = 1 TO 12
  READ d(i)
NEXT i
DIM day(1000) AS INTEGER
DIM mon(1000) AS INTEGER
DIM num(1000) AS INTEGER
DIM s AS STRING
LINE INPUT s
day1 = VAL(MID$(s, 1, 2))
mon1 = VAL(MID$(s, 4, 2))
INPUT N
k = 0
FOR j = 1 TO N
  LINE INPUT s
  c$ = MID$(s, 1, 4)
  day2 = VAL(MID$(s, 6, 2))
  mon2 = VAL(MID$(s, 9, 2))
  IF diff(day2, mon2, day1, mon1) > 3 THEN
    k = k + 1
    day(k) = day2
    mon(k) = mon2
    num(k) = VAL(c$)
  ENDIF
NEXT j
FOR i = 1 TO k - 1
  FOR j = 1 TO k - i
    IF (mon(j) > mon(j + 1)) OR
      ((mon(j) = mon(j + 1)) AND (day(j) > day(j + 1))) THEN
      day1 = day(j): mon1 = mon(j): num1 = num(j)
      day(j) = day(j + 1): mon(j) = mon(j + 1): num(j) = num(j + 1)
      day(j + 1) = day1: mon(j + 1) = mon1: num(j + 1) = num1
    END IF
  NEXT j
NEXT i

```

```

ENDIF
NEXT j
NEXT i
FOR i = 1 TO k
PRINT num(i)
NEXT i
END
FUNCTION diff (d1, m1, d2, m2)
IF m1 = m2 THEN diff = d2 - d1 ELSE diff = d2 + d(m1) - d1
END FUNCTION

```

Указания по оцениванию	Баллы
Программа работает верно и эффективно, т.е. корректно выделяет из входных данных даты, запоминает информацию только о тех ячейках, время хранения в которых истекло. Номера этих ячеек верно сортируются согласно времени сдачи багажа, а затем печатаются. Допускается наличие одной из нерациональностей, упомянутых в критериях присвоения 3-х баллов. Допускается наличие в тексте программы одной пунктуационной ошибки или опечатки	4
Программа работает в целом верно, но содержит по крайней мере две из следующих нерациональностей: сохраняются номера и даты для всех ячеек, сортируются все даты, а при печати анализируется, какие из них допустимые, каждый месяц анализируется отдельно на предмет количества дней в нем с помощью оператора варианта или условных операторов. 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, недопустимая для соответствующего типа данных	3
Программа работает в целом верно, но не всегда верно определяет искомые даты или некорректно работает в случае отсутствия искомого дат. Возможно, в реализации алгоритма содержатся 1–2 ошибки (используется знак "<" вместо ">", "or" вместо "and" и т.п.) или данные не всегда верно сравниваются при сортировке. 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, недопустимая для соответствующего типа данных.	2
Программа неверно работает при некоторых входных данных и, возможно, содержит ошибку в алгоритме сортировки, или даты сортируются верно, а соответствующие им номера ячеек – 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 языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных.	1
Задание выполнено неверно	0
<i>Максимальный балл</i>	4



# СОДЕРЖАНИЕ

АНАЛИЗ РЕЗУЛЬТАТОВ ЭКЗАМЕНА 2008 ГОДА ПО ИНФОРМАТИКЕ	4
МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЕГЭ 2009...	8
Общие замечания	8
1 Информационные процессы и системы	9
1.1. Информация и ее кодирование	9
1.2. Алгоритмизация и программирование	14
1.3. Основы логики	29
1.4. Моделирование	33
2 Информационные и коммуникационные технологии	35
2.1 Файловые системы	35
2.2. Обработка графической информации	36
2.3. Обработка информации в электронных таблицах	37
2.4 Базы данных	41
2.5 Телекоммуникационные технологии	42
2.6 Технология программирования	45
ТРЕНИРОВОЧНЫЕ ЗАДАНИЯ	50
1. Информационные процессы и системы	50
1.1. Информация и ее кодирование	50
1.2. Алгоритмизация и программирование	55
1.3. Основы логики	71
1.4 Моделирование	75
2 Информационные и коммуникационные технологии	78
2.1 Файловые системы	78
2.2. Обработка графической информации	79
2.3. Обработка информации в электронных таблицах	80
2.4. Базы данных	84
2.5. Телекоммуникационные технологии	86
2.6. Технологии программирования	88
ТРЕНИРОВОЧНЫЕ ВАРИАНТЫ 2009 Г.	93
Вариант 1	93
Вариант 2	101
ОТВЕТЫ И РЕШЕНИЯ	111
Ответы к тренировочным заданиям	111
Ответы к тренировочным вариантам-2009	122
Ответы к варианту 1	122
Ответы к варианту 2	128