



В

помощь
старшекласснику

Н. Н. Самыкина, А. П. Сильченко

Информатика



*все темы для
подготовки к ЕГЭ*

- Эффективная подготовка к урокам, ЕГЭ
- Повторение и обобщение материала
- Помощь при выполнении домашних заданий

Н. Н. Самыкина, А. П. Сильченко

Информатика

*все темы для
подготовки
к ЕГЭ*



В

помощь
старшекласснику

МОСКВА



ЭКСМО

2011

УДК 373.167.1:004
ББК 32.81я7
С 17

Самылкина Н. Н.

С 17 Информатика : все темы для подготовки к ЕГЭ / Н. Н. Самылкина, А. П. Сильченко. — М. : Эксмо, 2011. — 208 с. — (В помощь старшекласснику).

ISBN 978-5-699-42984-4

Издание окажет помощь старшеклассникам при подготовке к ЕГЭ по информатике. Книга поможет быстро и эффективно повторить и обобщить учебный материал. В пособие включен теоретический материал по основным темам ЕГЭ. Задания для самоконтроля помогут закрепить знания.

Издание подготовлено в соответствии с современными требованиями школьной программы среднего (полного) общего образования.

УДК 373.167.1:004

ББК 32.81я7

ISBN 978-5-699-42984-4

© Самылкина Н.Н., Сильченко А.П., 2011

© Оформление.

ООО «Издательство «Эксмо», 2011

Содержание

Введение	6
Раздел 1. ИНФОРМАЦИЯ И ЕЕ КОДИРОВАНИЕ	
1.1. Измерение количества информации	7
1.1.1. Понятие «информация»	7
1.1.2. Вероятностный метод измерения информации	8
1.1.3. Алфавитный метод измерения информации ..	11
1.2. Единицы измерения количества информации	11
1.3. Скорость передачи информации и пропускная способность канала связи	15
1.4. Комбинаторные задачи	17
1.5. Представление числовой информации	20
1.5.1. Позиционная система счисления	20
1.5.2. Формирование целых чисел в системах счисления	22
1.5.3. Перевод из произвольных систем счисления в десятичную	24
1.5.4. Перевод десятичных чисел в другие системы счисления	26
1.5.5. Перевод из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления, и наоборот	30
1.5.6. Арифметические операции в системах счисления	33
1.6. Кодирование информации	37
1.6.1. Кодирование целых чисел	38
1.6.2. Кодирование текстовой информации	39
1.6.3. Кодирование графической информации	40
<i>Задачи для самостоятельной работы</i>	43

Раздел 2. ОСНОВЫ ЛОГИКИ

2.1. Основные понятия логики	59
2.2. Логические операции	60
2.2.1. Отрицание «НЕ»	60
2.2.2. Конъюнкция «И»	61
2.2.3. Дизъюнкция «ИЛИ»	62
2.2.4. Импликация «ЕСЛИ – ТО»	63
2.2.5. Равносильность «ТОГДА И ТОЛЬКО ТОГДА»	64
2.2.6. Приоритет логических операций	65
2.3. Основные законы алгебры логики	67
2.4. Решение логических задач	78
2.4.1. Решение логических задач средствами алгебры логики	78
2.4.2. Решение логических задач табличным способом	80
2.4.3. Решение логических задач с помощью рассуждений	83
<i>Задачи для самостоятельной работы</i>	84

Раздел 3. ОСНОВЫ АЛГОРИТМИЗАЦИИ

3.1. Основные понятия	90
3.2. Словесный способ представления алгоритма	91
3.3. Графический способ представления алгоритма	101
3.4. Способ записи алгоритма в виде графа	107
<i>Задачи для самостоятельной работы</i>	113

Раздел 4. ПРОГРАММИРОВАНИЕ

4.1. Особенность языка Паскаль	124
4.2. Общая структура программ на языке Паскаль	125
4.3. Алфавит и идентификаторы	126
4.4. Описание переменных. Типы данных	127
4.5. Арифметические операции в Паскале	129
4.6. Операции отношения и логические операции	134
4.7. Основные операции языка Турбо-Паскаль	134
4.7.1. Оператор присваивания	134
4.7.2. Составной оператор	136
4.7.3. Операторы ввода-вывода	136
4.7.4. Условный оператор	139

4.7.5. Оператор варианта «CASE»	146
4.7.6. Комментарии	148
4.7.7. Оператор цикла	149
4.8. Массивы	160
4.8.1. Понятие о массиве	160
4.8.2. Одномерные массивы	161
4.8.3. Двумерные массивы	168
4.8.4. Полная переменная	172
4.9. Создание программ для решения прикладных задач	172
4.10. Отладка программы в среде Турбо-Паскаль	185
4.10.1. Основные правила написания текста программы	185
4.10.2. Стиль программирования, облегчающий отладку	186
4.10.3. Основные виды ошибок программирования	187
4.10.4. Средства, которыми располагает отладчик Турбо-Паскаля	189
<i>Задачи для самостоятельной работы</i>	191
Список литературы	204

ВВЕДЕНИЕ

Содержание измерителей по предмету «Информатика и ИКТ» и процедура контроля в едином государственном экзамене подразумевают проверку в основном теоретических знаний, а также вычислительных и алгоритмических умений учащихся, которые, в свою очередь, составляют основу предмета, и в том числе его технологической части. Поэтому авторы предлагают при подготовке к экзамену повторить основной теоретический материал с использованием данного издания и потренироваться в решении типовых задач.

Теоретическую часть предмета составляют темы, вошедшие в пособие: «Информация и ее кодирование», «Системы счисления», «Основы логики», «Основы алгоритмизации и программирования».

Указанные темы представлены достаточно подробно. Например, тема «Информация и ее кодирование» содержит вопросы измерения информации, способы кодирования разного вида информации, скорость передачи и пропускную способность каналов связи. Здесь же представлены различные способы решения задач: с использованием типовых формул, основ комбинаторики, графов и таблиц. Рассматриваемые задания расположены по увеличению их сложности.

Учебное пособие предназначено старшеклассникам для подготовки к ЕГЭ, учителям информатики и ИКТ для организации подготовки к экзамену, методистам по предмету.

Раздел 1

ИНФОРМАЦИЯ И ЕЕ КОДИРОВАНИЕ

1.1. ИЗМЕРЕНИЕ КОЛИЧЕСТВА ИНФОРМАЦИИ

1.1.1. Понятие «информация»

Количество информации, заключенное в сообщении, определяется объемом знаний, который несет это сообщение получающему человеку. Сообщение содержит информацию для человека, если заключенные в нем сведения являются для этого человека **новыми и понятными** и, следовательно, пополняют его знания.

Единица измерения количества информации называется бит. Сообщение, уменьшающее неопределенность знаний человека в два раза, несет для него 1 бит информации.

Бит — минимальная единица количества информации. Его можно представить как выбор ответа «да» или «нет» на поставленный вопрос. Если подбросить монету и проследить, какой стороной она упадет, то мы получим определенную информацию. Обе стороны монеты «равноправны», поэтому одинаково вероятно, что выпадет как одна, так и другая сторона. В таких случаях говорят, что свершившееся событие уменьшило неопределенность наших знаний о нем в 2 раза, следовательно, был получен 1 бит информации.

Если положить в мешок два шарика разного цвета, то, вытащив вслепую один шар, мы также получим информацию о цвете шара, равную 1 биту. В информатике обычно «да» обозначается цифрой 1, «нет» — цифрой 0. В логике двоичность осуществляется в паре «истина» — 1 и «ложь» — 0. Электронным представлением бита на

компьютере является ситуация «есть сигнал/нет сигнала».

Существует несколько методов измерения количества информации.

1.1.2. Вероятностный метод измерения информации

Давайте для начала разберемся с понятием «вероятность». Введем следующие понятия:

- испытание — любой эксперимент;
- единичное испытание — испытание, в котором совершается одно действие с одним предметом (например, подбрасывается монетка, или из корзины извлекается шар);
- исходы испытаний — результаты испытания (например, при подбрасывании монеты выпал «орел», или из корзины извлекли белый шар);
- множество исходов испытания — множество всех возможных исходов испытания;
- случайное событие — событие, которое может произойти или не произойти (например, выигрыш билета в лотерею, извлечение карты определенной масти из колоды карт).

Вероятностью случайного события (p) называется отношение числа благоприятствующих событию исходов (m) к общему числу исходов (n):

$$p = \frac{m}{n}. \quad (1)$$

Заметим, что вероятность случайного события может изменяться от 0 до 1.

Пример 1. В беспроигрышной лотерее разыгрывается 3 книги, 2 альбома, 10 наборов маркеров, 10 блокнотов. Какова вероятность выиграть книгу?

Решение. Общее число исходов $2 + 3 + 10 + 10 = 25$; число благоприятствующих исходу событий равно 3. Вероятность выигрыша книги вычисляется по формуле (1):

$$p = \frac{3}{25} = 0,12.$$

Верный ответ: 0,12.

Пусть в некотором сообщении содержатся сведения о том, что произошло одно из N **равновероятных** событий (равновероятность обозначает, что ни одно событие не имеет преимуществ перед другими). Тогда x бит — количество информации, заключенное в этом сообщении, и число N связаны формулой:

$$2^x = N. \quad (2)$$

Пример 2. В корзине лежат 8 шаров. Все шары разного цвета. Сколько информации содержит сообщение о том, что из корзины достали красный шар?

Решение. Поскольку вытаскивание любого из 8 шаров равновероятно, то количество информации о цвете вынутого шара находится из уравнения $2^x = 8$. Следовательно, $x = 3$ бита.

Верный ответ: 3 бита.

Пример 3. Шахматная доска состоит из 64 полей: 8 столбцов на 8 строк. Какое количество бит несет сообщение о выборе одного шахматного поля?

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

Решение. Поскольку выбор любой из 64 клеток равновероятен, то количество бит находится из формулы $2^x = 64$. Следовательно, $x = 6$ бит.

Верный ответ: 2).

Число равновероятных ситуаций (вариантов) далеко не всегда кратно 2, однако число бит может быть только целым (по простой причине, что полбита в компьютерной памяти нереализуемо). В данной ситуации ищется бли-

жайшая степень с основанием 2, превосходящая исходное число вариантов, показатель степени и есть искомое количество бит.

Пример 4. Для кодирования музыкального произведения используется 7 нотных знаков. Чему равен информационный объем произведения, состоящего из 180 нот?

- | | |
|------------|------------|
| 1) 180 бит | 3) 540 бит |
| 2) 360 бит | 4) 450 бит |

Решение. Одна нота выбирается из 7 вариантов, поэтому выбор одной ноты несет 3 бита информации: $2^x = 7 < 8 = 2^3$, $x = 3$ бита. Сообщение из 180 нот имеет объем $V = 180 \times 3$ бита = 540 бит.

Верный ответ: 3).

Заметим, что во многих случаях события происходят с **разной** вероятностью, а значит, формула (2) не всегда применима.

Связь между вероятностью события (p) и количеством информации (x) в сообщении об этом событии выражается формулой:

$$2^x = \frac{1}{p}. \quad (3)$$

Пример 5. В корзине лежат 8 черных шаров и 24 белых. Сколько бит информации несет сообщение о том, что достали черный шар?

- | | |
|------------|-----------|
| 1) 24 бита | 3) 8 бит |
| 2) 2 бита | 4) 4 бита |

Решение. Общее число исходов: $8 + 24 = 32$, число благоприятствующих исходу событий равно 8. Вероятность выбора черного шара определяется как $p = \frac{8}{32} = \frac{1}{4} = 0,25$.

Количество информации вычисляем из соотношения $2^x = \frac{1}{0,25} = \frac{1}{\frac{1}{4}} = 4$, значит, $x = 2$ бита.

Верный ответ: 2).

Качественная связь между вероятностью события и количеством информации в сообщении состоит в следующем: чем меньше вероятность некоторого события, тем больше информации содержит сообщение об этом событии.

1.1.3. Алфавитный метод измерения информации

Конечное множество различных символов (ноты, буквы, цифры), используемых при записи текста, называется **алфавитом**. Количество символов в алфавите называется **мощностью алфавита**.

Пример 6. Алфавит племени Ахо содержит X символов, а алфавит племени Юхо содержит в 4 раза больше символов. Племена обменялись приветствиями длиной по 100 символов каждое. Количество бит информации в приветствии племени Ахо обозначим N_1 , в приветствии племени Юхо — N_2 . Выберите верное утверждение.

1) $N_1 = 4 N_2$

3) $N_2 - N_1 = 400$

2) $N_2 = 4 N_1$

4) $N_2 - N_1 = 200$

Решение. Предположим, что $2^n = X$, то есть 1 символ алфавита племени Ахо содержит n бит информации, тогда $N_1 = 100n$. Относительно племени Юхо получаем $4X = 4 \times 2^n = 2^{n+2}$, тогда информационный объем сообщения племени оценивается следующим образом: $N_2 = 100(n + 2) = 100n + 200 = N_1 + 200$. В результате получаем $N_2 - N_1 = 200$.

Верный ответ: 4).

1.2. ЕДИНИЦЫ ИЗМЕРЕНИЯ КОЛИЧЕСТВА ИНФОРМАЦИИ

Бит как единица информации слишком мала, поэтому постоянно используется другая более крупная единица измерения информации — байт.

Байт — наименьшая адресуемая часть памяти компьютера, равная 8 битам, или 8-значному двоичному числу: **1 байт = 8 бит.**

Если бит в теории информации — количество информации, которое несет сообщение, то в вычислительной технике битом называют наименьшую «порцию» памяти компьютера, необходимую для хранения одного из разрядов «0» и «1», используемых для внутримашинного представления данных и команд.

Работая с информацией на современных компьютерах, следует знать следующие единицы, производные от байта, при составлении которых используется число $1024 = 2^{10}$.

Для измерения информации используются не только биты и байты, но и более крупные единицы:

1 Кбайт (килобайт) = 2^{10} байт = 1024 байт.

1 Мбайт (мегабайт) = 2^{10} Кбайт = 1024 Кбайт.

1 Гбайт (гигабайт) = 2^{10} Мбайт = 1024 Мбайт.

1 Тбайт (терабайт) = 2^{10} Гбайт = 1024 Гбайт.

1 Пбайт (петабайт) = 2^{10} Тбайт = 1024 Тбайт.

Заполним таблицу с коэффициентами перевода производных единиц от байта друг в друга. Например, $1 \text{ Мб} = 2^{10} \text{ Кб}$, $1 \text{ Кб} = 2^{-10} \text{ Мб}$.

	Байт	Кило- байт	Мега- байт	Гига- байт	Тера- байт	Пега- байт
Б	1	2^{-10}	2^{-20}	2^{-30}	2^{-40}	2^{-50}
Кб	2^{10}	1	2^{-10}	2^{-20}	2^{-30}	2^{-40}
Мб	2^{20}	2^{10}	1	2^{-10}	2^{-20}	2^{-30}
Гб	2^{30}	2^{20}	2^{10}	1	2^{-10}	2^{-20}
Тб	2^{40}	2^{30}	2^{20}	2^{10}	1	2^{-10}
Пб	2^{50}	2^{40}	2^{30}	2^{20}	2^{10}	1

Пример 7. Считая, что каждый символ кодируется одним байтом, оцените информационный объем следующего предложения: **Белеет Парус Одинокий В Тумане Моря Голубом!**

1) 352 бита

2) 44 бита

3) 352 байта

4) 88 байт

Решение. Так как в предложении 44 символа (считая знаки препинания и пробелы), то информационный объем вычисляется по формуле:

$$V = 44 \cdot 1 \text{ байт} = 44 \text{ байта} = 44 \cdot 8 \text{ бит} = 352 \text{ бита}.$$

Верный ответ: 1).

Пример 8. Объем сообщения равен 11 Кбайт. Сообщение содержит 11 264 символа. Какова мощность алфавита?

- 1) 64 2) 128 3) 256 4) 512

Решение. Выясним, какое количество бит выделено на 1 символ. Для этого переведем объем сообщения в биты ($11 \text{ Кбайт} = 11 \cdot 2^{10} \text{ байт} = 11 \cdot 2^{10} \cdot 2^3 \text{ бит} = 11 \cdot 2^{13} \text{ бит}$) и разделим его на число символов. На 1 символ приходится $\frac{11 \cdot 2^{13}}{11 \cdot 264} = \frac{11 \cdot 2^{13}}{11 \cdot 2^{10}} = 2^3 = 8 \text{ бит}$. Мощность алфавита опре-

деляем из формулы (1) $N = 2^8 = 256$ символов.

Верный ответ: 3).

Следующая задача немного сложнее задачи, рассмотренной в примере 3. Именно такого рода задачи предлагают решить участникам ЕГЭ по информатике в рамках данной темы.

Пример 9. Метеорологическая станция ведет наблюдение за влажностью воздуха. Результатом одного измерения является целое число от 0 до 100, которое записывается при помощи минимально возможного количества бит. Станция сделала 80 измерений. Определите информационный объем результатов наблюдений.

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

Решение. Любое значение влажности воздуха, выраженное в процентах, равновероятно. Количество информации о результате одного измерения влажности воздуха определяется из уравнения $2^x = 101$ (101 — количество

целых чисел в диапазоне от 0 до 100), или после поиска ближайшей степени числа 2 из неравенства $2^x = 128 = 2^7$. Таким образом, получаем $x = 7$ бит. Поскольку было сделано 80 измерений, то информационный объем сообщения равен

$$V = 80 \cdot 7 \text{ бит} = 560 \text{ бит} = 70 \text{ байт}.$$

Верный ответ: 2).

Следующий пример решается тем же способом, однако, чтобы не ошибиться в ответе, требуется внимательно читать задание. Советуем подчеркивать ключевые фразы задания.

Пример 10. В некоторой стране автомобильный номер длиной 6 символов составляется из заглавных букв (всего используется 19 букв) и десятичных цифр в любом порядке. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый номер — одинаковым и минимально возможным количеством байт. Определите объем памяти, необходимый для хранения 40 автомобильных номеров.

1) 120 байт

3) 200 байт

2) 160 байт

4) 240 байт

Решение. Каждый символ номера выбирается из 19 (буквы) + 10 (цифры) = 29 вариантов. Количество бит, необходимое для кодирования одного символа, определяется из уравнения $2^x = 29$, или после преобразования из уравнения $2^x = 32 = 2^5$. Таким образом, получаем $x = 5$ бит. Автомобильный номер состоит из 6 символов, следовательно, для хранения одного номера требуется $5 \cdot 6 = 30$ бит, при переводе в байты получаем

$$30 \text{ бит} < 32 \text{ бит} = 32 : 8 = 4 \text{ байта}.$$

Поскольку нужно определить объем памяти, отводимый для записи 40 номеров, умножая на 40, находим $V = 40 \cdot 4 \text{ байта} = 160 \text{ байт}$.

Верный ответ: 2).

1.3. СКОРОСТЬ ПЕРЕДАЧИ ИНФОРМАЦИИ И ПРОПУСКНАЯ СПОСОБНОСТЬ КАНАЛА СВЯЗИ

Общая схема передачи информации включает в себя отправителя информации, канал передачи информации и получателя информации.

Основной характеристикой каналов передачи информации является их пропускная способность (скорость передачи информации в ед. времени). Пропускная способность канала равна количеству информации, которое может передаваться по нему в единицу времени.

Объем переданной информации V вычисляется по формуле:

$$V = q \cdot t, \quad (4)$$

где q — пропускная способность канала (в битах в секунду или подобных единицах), а t — время передачи.

Обычно пропускная способность измеряется в битах в секунду (бит/с) и кратных единицах Кбит/с и Мбит/с. Однако иногда в качестве единицы используется байт в секунду (байт/с) и кратные ему единицы Кбайт/с и Мбайт/с.

Пример 11. Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать 100 страниц текста в 30 строк по 60 символов каждая, при условии, что каждый символ кодируется 1 байтом?

Решение. Вычислим объем файла в битах $V = 100 \times 30 \times 60 \times 8$ бит = 1 440 000 бит. Скорость передачи сообщения $q = 28\,800$ бит/с. Время равно

$$t = \frac{V}{q} = \frac{1\,440\,000}{28\,800} = 50 \text{ секунд.}$$

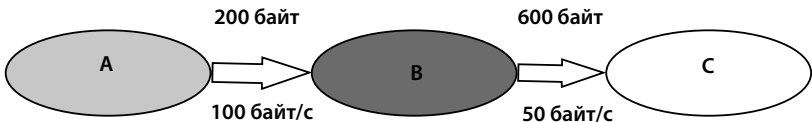
Верный ответ: 50.

Рассмотрим более сложную задачу.

Пример 12. Устройство A передает информацию устройству C через устройство B в рамках следующих правил:

1. Информация передается пакетами по 200 байт.
2. Устройство B может одновременно принимать информацию от устройства A и передавать ранее полученную информацию устройству C .
3. Устройство B может передавать очередной пакет устройству C только после того, как полностью получит этот пакет от устройства A .
4. Устройство B обладает неограниченным по объему буфером, в котором может хранить полученные от устройства A , но еще не переданные устройству C пакеты. Пропускная способность канала между A и B — 100 байт в секунду. Пропускная способность канала между B и C — 50 байт в секунду. Было отправлено два пакета информации. Через сколько секунд C закончит прием всей информации от A ? В ответе укажите целое число.

Решение. Так как скорость приема информации устройством B больше, чем скорость ее передачи устройству C , то время передачи сложится из двух этапов. Продемонстрируем это графически.



Время передачи первого пакета информации от A устройству B равно $t_1 = \frac{V_1}{q_1} = \frac{200}{100} = 2$ секунды. Далее

прием информации от A и передача ее устройству C осуществляются устройством B одновременно, поэтому достаточно вычислить время передачи всех трех пакетов ин-

формации от B к C $t_2 = \frac{V_2}{q^2} = \frac{600}{50} = 12$ секунд. Общее время передачи $t = t_1 + t_2 = 2 + 12 = 14$ секунд.

Верный ответ: 14.

1.4. КОМБИНАТОРНЫЕ ЗАДАЧИ

К данному разделу относятся задачи **В1** и **С3**.

Комбинаторика — это раздел математики, связанный с изучением количества различных комбинаций, которые можно составить из заданных объектов.

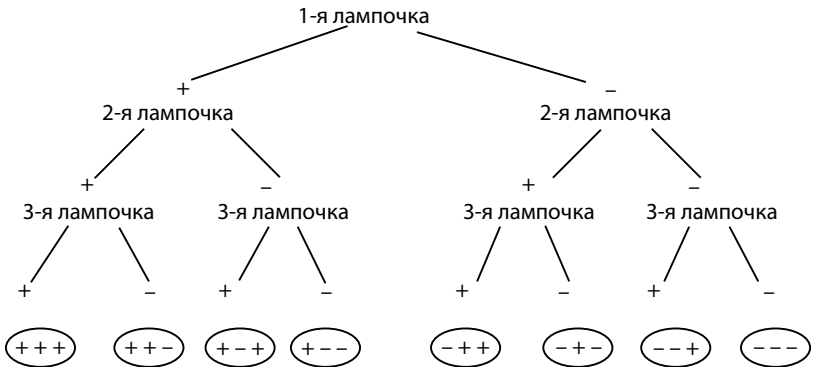
Пусть дано множество, состоящее из n различных элементов (например, {белый (Б), синий (С), красный (К)}). Составим из элементов этого множества комбинации, содержащие k элементов. Эти комбинации могут отличаться друг от друга как составом элементов, так и порядком их расположения (например, БСКС, СКСС — 2 комбинации из 4 элементов). Комбинации, построенные по этим правилам, называются *выборки*. Число различных выборок, содержащих k элементов (элементы могут повторяться), выбираемых из исходного множества, включающего n различных элементов, определяется по формуле:

$$N = n^k. \quad (5)$$

Для того чтобы не запутаться в различных вариантах комбинаций и не пропустить ни одной из них, удобно использовать наглядное представление в виде графа. **Граф**, или **дерево вариантов**, — это множество вершин и множество ребер, которые соединяют не более двух вершин.

Пример 13. В коридоре три лампочки. Сколько имеется различных способов освещения коридора (включая случай, когда все лампочки не горят)?

Решение. Обозначим горящую лампочку знаком «+», а негорящую — знаком «-». Построим дерево вариантов (см. рис. на с. 18).



В результате получили 8 способов освещения. Эту же задачу можно решить гораздо быстрее, используя понятие выборки. Число различных элементов в исходном множестве $n = 2$ («+» и «-»). Число элементов в выборке $k = 3$ (3 лампочки). Общее число вариантов освещения определяется формулой $V = 2^3 = 8$.

Верный ответ: 8.

Пример 14. Укажите минимальное число символов в алфавите, чтобы с помощью слов из пяти букв можно было бы передавать 220 различных сообщений. Слова могут содержать повторяющиеся символы.

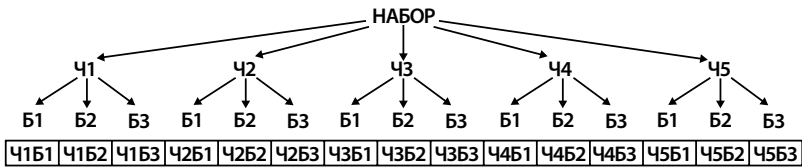
Решение. Из формулы 5 получаем неравенство $x^5 \geq 220$. Подбором ($2^5 = 32 < 220$, $3^5 = 243 > 220$) получаем $x = 3$.

Верный ответ: 3.

Пример 15. В магазине «Все для чая» есть 5 разных чашек и 3 разных блюдца. Сколькими способами можно купить чашку с блюдцем?

Решение. Так как чашки и блюдца выбирают из разных множеств, то формула (5) для решения задачи не применима. Предлагаем следующие способы решения.

Способ 1. Если строить дерево вариантов, то получаем граф:



Пересчитав комбинации в последней строке, получаем 15 наборов.

Способ 2. Для каждой чашки есть 3 варианта выбора блюда, а так как чашку можно выбрать 5 способами, то получаем $5 \times 3 = 15$ наборов.

Верный ответ: 15.

Если элементы выборки не повторяются, но порядок элементов имеет значение, то число различных выборов из r элементов, выбираемых из исходного множества из n элементов, вычисляется по формуле:

$$A_n^k = n(n-1)(n-2) \cdot \dots \cdot (n-r+1). \quad (6)$$

A_n^k указывает число различных размещений из n элементов по r позициям.

Если $n = r$, то различные выборки отличаются друг от друга только порядком элементов. Такие выборки называются перестановками из n элементов. Число различных перестановок:

$$P_n = n(n-1) \cdot \dots \cdot 2 \cdot 1 = n! \quad (7)$$

Произведение подряд идущих первых n натуральных чисел обозначают $n!$ и называют «эн факториал».

Пример 16. Сколько существует двузначных чисел, в записи которых цифры 1, 2, 3, 4, 5 встречаются ровно по одному разу?

Решение.

Способ 1. Если строить дерево вариантов, то получаем таблицу А (см. с. 21).

В итоге получаем 20 чисел.

Способ 2. Первую цифру можем выбрать 5 способами. Для каждой первой цифры вторая выбирается 4 способами, так как вторая цифра не может совпадать с первой. В результате получаем $A_5^2 = 5 \times 4 = 20$ чисел.

Верный ответ: 20.

Пример 17. Сколькими способами 4 человека могут по одному разбежаться на все 4 стороны.

Решение. Построим дерево вариантов в виде таблицы (см. табл. Б на с. 21).

Видим, что первый человек может выбирать между 4 сторонами света, второй человек выбирает из оставшихся 3 сторон света. У третьего человека остается выбор из двух вариантов, а у четвертого человека выбора нет, ему остался единственный вариант. В результате получаем $P_4 = 4! = 4 \times 3 \times 2 \times 1 = 24$ способа.

Верный ответ: 24.

1.5. ПРЕДСТАВЛЕНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ

1.5.1. Позиционная система счисления

Система счисления — это способ представления чисел и соответствующие ему правила действий над числами. Знаки, используемые при записи чисел, называются цифрами. Все цифры называются алфавитом системы счисления. Количество используемых цифр называется основанием системы счисления.

Система счисления, применяемая в современной математике, является позиционной десятичной системой. Ее основание равно 10, так как запись любых чисел производится с помощью десяти цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Таблица А

1-я цифра	1				2				3				1-я цифра				1-я цифра										
	2	3	4	5	1	3	4	5	1	2	4	5	1	2	3	5	1	2	3	5	2-я цифра	2-я цифра	Число	Число	51	52	53
2-я цифра																											
Число	12	13	14	15	21	23	24	25	31	32	34	35	41	42	43	45	51	52	53	54							

Таблица Б

1-й человек	С						Ю						З						В																	
	Ю			3			В			С			З			Ю			В			С			Ю			З								
2-й человек	3						В						С						Ю						З											
3-й человек	3	В	Ю	В	Ю	3	3	В	С	В	С	3	3	В	С	В	С	3	3	В	С	В	С	3	3	В	С	В	С	3	3	В	С	В	С	3
4-й человек	В	3	В	Ю	3	Ю	В	3	Ю	В	3	С	В	3	Ю	В	3	С	В	3	Ю	В	3	С	В	3	Ю	В	3	С	В	3	Ю	В	3	С

Однако развитие информатики привело к активному использованию систем счисления с основаниями, кратными числу 2.

Основание	Название	Алфавит
$n = 2$	двоичная	0 1
$n = 8$	восьмеричная	0 1 2 3 4 5 6 7
$n = 16$	шестнадцатеричная	0 1 2 3 4 5 6 7 8 9 A B C D E F

Основанием позиционной системы счисления может быть любое натуральное число (например, 5, 21, 37). Во избежание путаницы справа от числа нижним индексом приписывают основание: 101101_2 , 367_8 , $3B8A_{16}$, $3AO_{37}$.

1.5.2. Формирование целых чисел в системах счисления

В каждой системе счисления цифры упорядочены в соответствии с их значениями: 1 больше 0, 2 больше 1 и т.д. **Продвижением цифры** называют замену ее следующей по величине. Продвинуть цифру 1 — значит заменить ее на 2, продвинуть цифру 2 — значит заменить ее на 3 и т.д. Продвижение старшей цифры (например, цифры 9 в десятичной системе) означает замену ее на 0. В двоичной системе, использующей только 2 цифры — 0 и 1, продвижение 0 означает замену его на 1, а продвижение 1 — замену ее на 0.

Целые числа в любой системе счисления порождаются с помощью **Правила счета**: для образования целого числа, следующего за любым данным целым числом, нужно продвинуть самую правую цифру числа; если какая-либо цифра после продвижения стала нулем, то нужно продвинуть цифру, стоящую слева от нее.

Применяя это правило, запишем первые десять целых чисел

в двоичной системе:

0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001;

в троичной системе:

0, 1, 2, 10, 11, 12, 20, 21, 22, 100;

в пятеричной системе:

0, 1, 2, 3, 4, 10, 11, 12, 13, 14;

в восьмеричной системе:

0, 1, 2, 3, 4, 5, 6, 7, 10, 11.

Следует запомнить запись в системах счисления первых двух десятков целых чисел:

Таблица 1

A_{10}	0	1	2	3	4	5	6
A_2	0	1	10	11	100	101	110
A_8	0	1	2	3	4	5	6
A_{16}	0	1	2	3	4	5	6
A_{10}	7	8	9	10	11	12	13
A_2	111	1000	1001	1010	1011	1100	1101
A_8	7	10	11	12	13	14	15
A_{16}	7	8	9	A	B	C	D
A_{10}	14	15	16	17	18	19	20
A_2	1110	1111	10000	10001	10010	10011	10100
A_8	16	17	20	21	22	23	24
A_{16}	E	F	10	11	12	13	14

Развернутой формой записи числа называется запись в виде

$$A_n = \pm (a_{m-1}n^{m-1} + a_{m-2}n^{m-2} + \dots + a_0n^0 + a_{-1}n^{-1} + a_{-2}n^{-2} + \dots + a_{-k}n^{-k}).$$

Здесь A_n — само число, n — основание системы счисления, a_i — цифры данной системы счисления, m — количество разрядов целой части числа, k — количество разрядов дробной части числа.

Так, например,

$$765,345_{10} = 7 \times 10^2 + 6 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2} + 5 \times 10^{-3},$$

$$1011,01_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2},$$

$$10FC_{16} = 1 \times 16^3 + 0 \times 16^2 + F \times 16^1 + C \times 16^0.$$

Обратите внимание, что при подсчете выражения получится десятичный эквивалент числа.

1.5.3. Перевод из произвольных систем счисления в десятичную

Если все слагаемые в развернутой форме недесятичного числа представить в десятичной системе и вычислить полученное выражение по правилам десятичной арифметики, то получится число в десятичной системе, равное данному. По этому принципу производится перевод чисел из недесятичной системы в десятичную систему счисления.

Пример 18. Заданы четыре числа в различных системах счисления: $A = 232_4$, $B = 2F_{16}$, $C = 53_8$, $D = 101100_2$. Из максимального из этих чисел вычли минимальное. Напишите, что получилось в результате, переведя результат в десятичную систему счисления.

Решение. Переведем все числа в десятичную систему счисления:

$$A = 2 \cdot 4^2 + 3 \cdot 4^1 + 2 = 46_{10}, \quad B = 1 \cdot 16^1 + 15 = 47_{10},$$

$$C = 5 \cdot 8^1 + 3 = 43_{10}, \quad D = 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 = 44_{10}.$$

Видим, что максимальным является число B , а минимальным — C . Разность между максимальным и минимальным равна $B - C = 47 - 43 = 4$.

Верный ответ: 4.

Пример 19. Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 17, запись которых в системе счисления с основанием три оканчивается на две одинаковые цифры.

Решение. Напоминаем, что в троичной системе счисления присутствуют цифры 0, 1, 2. Будем последовательно выписывать числа, оканчивающиеся на две одинаковые цифры, и осуществлять перевод:

$$11_3 = 1 \cdot 3^1 + 1 = 4_{10},$$

$$22_3 = 2 \cdot 3^1 + 2 = 8_{10},$$

$$100_3 = 1 \cdot 3^2 + 0 \cdot 3^1 + 0 = 9 + 0 + 0 = 9_{10},$$

$$111_3 = 1 \cdot 3^2 + 1 \cdot 3^1 + 1 = 9 + 3 + 1 = 13_{10},$$

$$122_3 = 1 \cdot 3^2 + 2 \cdot 3^1 + 2 = 9 + 6 + 2 = 17_{10}.$$

Понятно, что следующее число 200_3 будет число уже больше 17.

Верный ответ: 4, 8, 9, 17.

Пример 20. В системе счисления с некоторым основанием число 129 записывается в виде 1004. Укажите это основание.

Решение. Обозначим неизвестное основание через x и запишем формулу перевода:

$$129_{10} = 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 4 = x^3 + 4.$$

Решая уравнение $x^3 + 4 = 129$, получаем $x = 5$.

Верный ответ: 5.

Пример 21. Укажите основание позиционной системы счисления x , в которой будет справедливо следующее равенство: $13x + 31x = 110x$.

Решение. Запишем формулу перевода

$$1 \cdot x^1 + 3 + 3 \cdot x^1 + 1 = 1 \cdot x^2 + 1 \cdot x^1 + 0.$$

Решая уравнение $4x + 4 = x^2 + x$, получаем $x = 4$ ($x = -1$ является посторонним корнем).

Верный ответ: 4.

Пример 22. Чему равно наименьшее основание позиционной системы счисления Y , при котором $225x = 14y$?

Решение. Запишем формулу перевода

$$2 \cdot x^2 + 2 \cdot x^1 + 5 = 1 \cdot y^1 + 4,$$

$$\text{или } y = 2x^2 + 2x + 1.$$

Заметим, что в числе с основанием x присутствует цифра 5, значит, $x \geq 6$. Подставляя вместо x число 6, находим наименьшее из возможных $y = 2 \cdot 6^2 + 2 \cdot 6 + 1 = 85$.

Верный ответ: 85.

1.5.4. Перевод десятичных чисел в другие системы счисления

Перевод целых чисел (алгоритм 1):

1) основание новой системы счисления выразить в десятичной системе счисления и все последующие действия производить в десятичной системе счисления;

2) последовательно выполнять деление данного числа и получаемых неполных частных на основание новой системы счисления до тех пор, пока не получим неполное частное меньше делителя;

3) полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления;

4) составить число, записывая его, начиная с последнего частного.

Пример 23. Переведем число 75 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную.

в двоичную

$$\begin{array}{r}
 75 \overline{) 2} \\
 \underline{74} \quad 37 \overline{) 2} \\
 1 \quad \underline{36} \quad 18 \overline{) 2} \\
 \quad \quad \underline{1} \quad 18 \quad 9 \overline{) 2} \\
 \quad \quad \quad \quad \underline{0} \quad 8 \quad 4 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \underline{1} \quad 4 \quad 2 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad \underline{0} \quad 2 \quad 1 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad \quad \underline{0} \quad 0 \quad 0 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \underline{1} \quad 0 \quad 1
 \end{array}$$

в восьмеричную

$$\begin{array}{r}
 75 \overline{) 8} \\
 \underline{72} \quad 9 \overline{) 8} \\
 3 \quad \underline{8} \quad 1 \overline{) 8} \\
 \quad \quad \underline{1} \quad 0 \quad 0 \overline{) 8} \\
 \quad \quad \quad \quad \underline{1}
 \end{array}$$

в шестнадцатеричную

$$\begin{array}{r}
 75 \overline{) 16} \\
 \underline{64} \quad 4 \overline{) 16} \\
 11 \quad \underline{0} \quad 0 \\
 \quad \quad \quad \underline{4}
 \end{array}$$

Замечание: остаток 11_{10} записывается шестнадцатеричной цифрой D_{16} .

$$\text{Ответ: } 75_{10} = 1001011_2 = 113_8 = 4B_{16}.$$

Перевод десятичной дроби (алгоритм 2):

Для перевода правильной десятичной дроби F в систему счисления с основанием q необходимо:

- 1) F умножить на q , записанное в десятичной системе,
- 2) дробную часть полученного произведения снова умножить на q ,

3) и т.д., до тех пор, пока дробная часть очередного произведения не станет равной нулю либо не будет достигнута требуемая точность изображения числа F в q -ичной системе,

4) представлением дробной части числа F в новой системе счисления будет последовательность целых частей полученных произведений, записанных в порядке их получения и изображенных одной q -ичной цифрой.

Пример 24. Переведем число 0,8125 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную:

$$\begin{array}{r} \textcircled{0,} \times \begin{array}{r} 8125 \\ \hline 2 \\ \hline 1 \quad 625 \\ \times \quad 2 \\ \hline 1 \quad 25 \\ \times \quad 2 \\ \hline 0 \quad 5 \\ \times \quad 2 \\ \hline 1 \end{array} \end{array}$$

$$\begin{array}{r} \textcircled{0,} \times \begin{array}{r} 8125 \\ \hline 8 \\ \hline 6 \quad 5 \\ \times \quad 8 \\ \hline 4 \quad 0 \end{array} \end{array}$$

$$\begin{array}{r} \textcircled{0,} \times \begin{array}{r} 8125 \\ \hline 16 \\ \hline \textcircled{D_{16}} \quad 13 \quad 0 \end{array} \end{array}$$

Замечание: число 13_{10} записывается шестнадцатеричной цифрой D_{16} .

Ответ: $0,8125_{10} = 0,1101_2 = 0,64_8 = 0, D_{16}$.

Для чисел, имеющих как целую, так и дробную части, перевод из десятичной системы счисления в другую осуществляется отдельно для целой и дробной частей по правилам, указанным выше.

Пример 25. Сколько единиц в двоичной записи десятичного числа 194,125?

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

Решение. Переведем в двоичную систему счисления числа 194 и 0,125:

$$\begin{array}{r} \begin{array}{r} -194 \quad | \quad 2 \\ \hline -194 \quad -97 \quad | \quad 2 \\ \hline 0 \quad -96 \quad -48 \quad | \quad 2 \\ \hline \quad 1 \quad -48 \quad -24 \quad | \quad 2 \\ \hline \quad \quad 0 \quad -24 \quad -4 \quad | \quad 2 \\ \hline \quad \quad \quad 0 \quad -4 \quad -12 \quad | \quad 2 \\ \hline \quad \quad \quad \quad 0 \quad -12 \quad -6 \quad | \quad 2 \\ \hline \quad \quad \quad \quad \quad 0 \quad -6 \quad -3 \quad | \quad 2 \\ \hline \quad \quad \quad \quad \quad \quad 0 \quad -2 \quad -1 \quad | \quad 2 \\ \hline \quad \quad \quad \quad \quad \quad \quad 1 \quad -0 \quad 0 \end{array} \end{array}$$

$$\begin{array}{r} \textcircled{0,} \times \begin{array}{r} 125 \\ \hline 2 \\ \hline 0 \quad 25 \\ \times \quad 2 \\ \hline 0 \quad 5 \\ \times \quad 2 \\ \hline 1 \quad 0 \end{array} \end{array}$$

В результате получаем $194,125_{10} = 11000010,001_2$.
В двоичной записи числа 194,125 четыре единицы.

Верный ответ: 4).

Пример 26. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 29 оканчивается на 5.

Решение.

1) Так как в записи числа присутствует цифра 5, то основание $x \geq 6$. Очевидно, что $x < 29$.

2) При переводе десятичного числа в другую систему счисления используется алгоритм 1, основанный на делении углом:

$$\begin{array}{r} 29 \mid x \\ \dots \mid \dots \\ 5 \end{array}$$

Значит, число $29 - 5 = 24$ делится на x без остатка. То есть делители числа 24 (2, 3, 4, 6, 8, 12, 24) и могут являться искомыми основаниями систем счисления.

3) Основания 2, 3 и 4 не подходят, так как мы выяснили, что $6 \leq x < 29$.

Верный ответ: 6, 8, 12, 24.

Рассмотрим очень похожую задачу, однако имеющую очень серьезную особенность.

Пример 27. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 63 оканчивается на 23.

Решение.

Так как в записи числа присутствует цифра 3, то основание $x \geq 4$. Очевидно, что $x < 63$.

При переводе десятичного числа в другую систему счисления используется алгоритм 1, основанный на делении углом:

$$\begin{array}{r} 63 \mid x \\ \dots \mid \dots \\ 3 \mid \dots \\ \dots \\ 2 \end{array}$$

Видим, что число $63 - 3 = 60$ делится на x без остатка. То есть среди делителей числа 60 (2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60) и будем искать основания систем счисления.

Основания 2 и 3 не подходят, так как мы выяснили, что $4 \leq x < 63$. Основание 10 дает нам исходное число, поэтому не будем его рассматривать.

Остальные делители придется проверять непосредственно делением.

$$\begin{array}{r|l} 63 & 4 \\ \hline 60 & 15 \\ \hline 3 & 12 \\ \hline & 3 \end{array}$$

$$\begin{array}{r|l} 63 & 5 \\ \hline 60 & 12 \\ \hline 3 & 10 \\ \hline & 2 \end{array}$$

$$\begin{array}{r|l} 63 & 6 \\ \hline 60 & 10 \\ \hline 3 & 6 \\ \hline & 4 \end{array}$$

$$\begin{array}{r|l} 63 & 12 \\ \hline 60 & 5 \\ \hline 3 & \end{array}$$

$$\begin{array}{r|l} 63 & 15 \\ \hline 60 & 4 \\ \hline 3 & \end{array}$$

$$\begin{array}{r|l} 63 & 20 \\ \hline 60 & 3 \\ \hline 3 & \end{array}$$

$$\begin{array}{r|l} 63 & 30 \\ \hline 60 & 2 \\ \hline 3 & \end{array}$$

$$\begin{array}{r|l} 63 & 60 \\ \hline 60 & 1 \\ \hline 3 & \end{array}$$

Таким образом, $63_{10} = 333_4 = 223_5 = 143_6 = 53_{12} = 43_{15} = 33_{20} = 23_{30} = 13_{60}$.

На 23 заканчиваются только числа с основаниями 5 и 30.

Верный ответ: 5, 30.

1.5.5. Перевод из двоичной системы счисления в восьмеричную и шестнадцатеричную системы счисления, и наоборот

Двоичная система, удобная для компьютеров, для человека неудобна из-за ее громоздкости. Перевод чисел из десятичной системы в двоичную систему счисления, и наоборот, выполняет компьютер.

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему очень прост: достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр). При этом удобно пользоваться таблицей 1.

Пример 28. Как записывается число 567_8 в двоичной системе счисления?

1) 1011101_2

3) 101110111_2

2) 100110111_2

4) 11110111_2

$$\begin{array}{ccccccc}
 A87_{16} = & A & 8 & 7 & & 101 & 010 & 000 & 111 \\
 & \downarrow & \downarrow & \downarrow & \Rightarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 & 1010 & 1000 & 0111 & & 5 & 2 & 0 & 7
 \end{array}$$

Верный ответ: 3).

Пример 31. Дано: $a = D7_{16}$, $b = 331_8$. Какое из чисел c , записанных в двоичной системе, отвечает условию $a < c < b$?

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

Решение. Переведем числа a и b в двоичную систему счисления.

$$\begin{array}{ccccccc}
 a_{16} = & D & 7 & b_8 & & b_8 = & 3 & 3 & 1 \\
 & \downarrow & \downarrow & \downarrow & & & \downarrow & \downarrow & \downarrow \\
 a_2 = & 1001 & 0111, & 0111 & & b_2 = & 011 & 011 & 001
 \end{array}$$

Между a и b находится только одно число 11011000.

$$11010111 < 1101000 < 11011001.$$

Верный ответ: 4).

Пример 32. Для передачи по каналу связи сообщения, состоящего только из символов А, Б, В и Г, используется посимвольное кодирование: А-00, Б-11, В-010, Г-011. Через канал связи передается сообщение: ВАГБГВ. Закодируйте сообщение данным кодом. Полученную двоичную последовательность переведите в шестнадцатеричный вид.

- 1) AD34 3) 101334
 2) 43DA 4) CADBCD

Решение. Переведем сообщение ВАГБГВ в двоичное число, используя таблицу:

А	Б	В	Г
00	11	010	011

$$ВАГБГВ \rightarrow 010\ 00\ 011\ 11\ 011\ 010.$$

Число 0100001111011010 переведем в шестнадцатеричную систему счисления (перевод тетрадами):

$$\begin{array}{cccc}
 0100 & 1011 & 1101 & 110_2 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 4 & 3 & D & A.
 \end{array}$$

Верный ответ: 2).

1.5.6. Арифметические операции в системах счисления

Рассмотрим арифметические операции сложения, вычитания, умножения и деления. Правила выполнения этих операций в десятичной системе хорошо известны — это сложение, вычитание, умножение столбиком и деление уголком. Эти правила применимы и ко всем другим позиционным системам счисления. Только таблицами сложения и умножения надо пользоваться особыми для каждой системы.

Сложение

Таблицы сложения в любой позиционной системе счисления легко составить, используя Правило счета.

Таблица 2

Сложение в двоичной системе			Сложение в восьмеричной системе								
+	0	1	+	0	1	2	3	4	5	6	7
0	0	1	0	0	1	2	3	4	5	6	7
1	1	10	1	1	2	3	4	5	6	7	10
0	0	1	2	3	4	5	6	7	10	11	12
1	1	10	3	4	5	6	7	10	11	12	13
			4	5	6	7	10	11	12	13	14
			5	6	7	10	11	12	13	14	15
			6	7	10	11	12	13	14	15	16
			7	10	11	12	13	14	15	16	

При сложении цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево.

Пример 33. Сложим числа 15 и 6 в различных системах счисления.

Решение. Переведем числа 15 и 6 в двоичную и восьмеричную системы счисления и выполним сложение, используя таблицу 2.

Десятичная:	Двоичная:	Восьмеричная:
$15_{10} + 6_{10}$	$11111_2 + 110_2$	$17_8 + 6_8$
$\begin{array}{r} 1 \\ 15 \\ + 6 \\ \hline 21 \end{array}$ <div style="margin-left: 40px;"> \swarrow $5+6=11=10+1$ \swarrow $1+1=2$ </div>	$\begin{array}{r} 111 \\ + 1111 \\ \hline 0110 \\ 10101 \end{array}$ <div style="margin-left: 40px;"> \swarrow $1+0=1$ \swarrow $1+1=2=10$ \swarrow $1+1+1=3=11$ \swarrow $1+1=2=10$ </div>	$\begin{array}{r} 1 \\ 17 \\ + 6 \\ \hline 25 \end{array}$ <div style="margin-left: 40px;"> \swarrow $7+6=13=8+5$ \swarrow $1+1=2$ </div>

Ответ: $15+6 = 21_{10} = 10101_2 = 25_8 = 15_{16}$.

Пример 34. Вычислите сумму чисел x и y , при $x = A6_{16}$, $y = 75_8$. Результат представьте в двоичной системе счисления.

1) 11011011_2

3) 11100011_2

2) 11110001_2

4) 10010011_2

Решение. Переведем числа x и y в двоичную систему счисления

$x =$	A	6	$y =$	7	5
	\downarrow	\downarrow		\downarrow	\downarrow
$x =$	1010	0110	$y =$	111	101

Пользуясь правилами сложения в двоичной системе счисления, получаем

$$\begin{array}{r} 1111 \\ + 10100110 \\ \hline 111101 \\ 11100011 \end{array}$$

\swarrow $1+0=1$
 \swarrow $1+0=1$
 \swarrow $1+1=10$
 \swarrow $1+1=10$
 \swarrow $1+1=10$
 \swarrow $1+1+1=10+1=11$

Верный ответ: 3).

Пример 35. Чему равна сумма чисел 43_8 и 56_{16} ?

- 1) 121_8
- 2) 171_8
- 3) 69_{16}
- 4) 1000001_2

Решение. Переведем число a_2 в восьмеричную систему счисления:

$56_{16} =$	5	6	$a_2 =$	1	010	110
	\downarrow	\downarrow		\downarrow	\downarrow	\downarrow
$x =$	101	0110	$a_8 =$	1	2	6

Пользуясь правилами сложения в восьмеричной системе счисления, получаем

$$\begin{array}{r} 1 \\ + 43_8 \\ + 126_8 \\ \hline 171_8 \end{array}$$

└ 3 + 6 = 9 = 8 + 1

Верный ответ: 2).

В вариантах ЕГЭ встречаются задачи и на вычитание.

Пример 36. Вычислите разность $X - Y$ двоичных чисел, если $X = 1010100_2$ и $Y = 1000010_2$. Результат представьте в двоичном виде.

- 1) 11010_2
- 2) 10100_2
- 3) 10010_2
- 4) 10101_2

Решение. Вычитание осуществляется по тем же правилам, что и в десятичной системе счисления.

$$\begin{array}{r} 1010100 \\ - 1000010 \\ \hline \end{array}$$

$1-0=0$
 $10-0=1$
 $0-0=0$
 $0-0=0$
 $1-0=0$
 $0-0=0$
 $1-1=0$

Верный ответ: 2).

Замечание. Если вам трудно складывать или вычитать в системах счисления, отличных от десятичной, можете перевести числа в десятичную систему счисления, выполнить арифметические действия, а затем результат перевести в требуемую в ответе систему счисления.

Умножение

Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо заимствовать из соответствующих рассматриваемой системе таблиц умножения и сложения.

Таблица 3

Умножение в двоичной системе			Умножение в восьмеричной системе								
*	0	1	*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	2	3	4	5	6	7
2	0	2	2	0	2	4	6	10	12	14	16
3	0	3	3	0	3	6	11	14	17	22	25
4	0	4	4	0	4	10	14	20	24	12	13
5	0	5	5	0	5	12	17	24	31	36	43
6	0	6	6	0	6	14	22	30	36	44	52
7	0	7	7	0	7	11	25	34	43	52	61

Пример 37. Перемножим числа 15 и 12.

Десятичная

$$15_{10} \cdot 12_{10}$$

$$\begin{array}{r} 15 \\ \times 12 \\ + 30 \\ \hline 15 \\ \hline 180 \end{array}$$

Двоичная

$$1111_2 \cdot 1100_2$$

$$\begin{array}{r} 1111 \\ \times 1100 \\ \hline 1111 \\ + 1111 \\ \hline 1111 \\ \hline 10110100 \end{array}$$

Восьмеричная:

$$18_8 \cdot 14_8$$

$$\begin{array}{r} 17 \\ \times 14 \\ + 74 \\ \hline 17 \\ \hline 264 \end{array}$$

Ответ: $15 \cdot 12 = 180_{10} = 10110100_2 = 264_8$.

Операция деления не отличается в разных системах счисления, следует только пользоваться грамотно теми цифрами, которые входят в алфавит используемой системы счисления.

1.6. КОДИРОВАНИЕ ИНФОРМАЦИИ

Код — это правило отображения одного набора объектов или знаков в другой набор знаков без потери информации. Чтобы избежать потерь информации, это отображение должно быть таким, чтобы можно было всегда однозначно возвратиться к прежнему набору объектов или знаков.

Например, любую информацию можно передать русским языком с помощью 33 букв русского алфавита и добавочных знаков препинания.

Кодирование — это представление, моделирование одного набора знаков другим с помощью кода. Кодовая таблица — это соответствие между набором знаков и их кодами, обычно разными числами.

При кодировании нет секретного ключа, как при шифровании, так как кодирование ставит целью лишь более сжатое, компактное представление сообщения.

Компьютер может обрабатывать только информацию, представленную в числовой форме. Вся другая информация (например, звуки, изображения, показания приборов и т. д.) для обработки на компьютере должна быть преобразована в числовую форму. При вводе в компьютер текстовой информации каждая буква кодируется определенным числом, а при выводе на внешние устройства (экран или печать) для восприятия человеком по этим числам строятся изображения букв. Соответствие между набором букв и числами называется **кодировкой символов**.

В компьютерах используют двоичную систему потому, что она имеет ряд преимуществ перед другими:

- для ее реализации используются технические элементы с двумя возможными состояниями (есть ток — нет тока);
- представление информации посредством только двух состояний надежно и помехоустойчиво;
- двоичная арифметика проще десятичной (двоичные таблицы сложения и умножения предельно просты).

1.6.1. Кодирование целых чисел

Множество целых чисел, представимых в памяти компьютера, ограничено. Диапазон зависит от размера ячеек памяти, используемых для их хранения. В k -разрядной ячейке может храниться 2^k различных значений целых чисел. Чтобы получить внутреннее представление такого числа, необходимо:

- перевести число в двоичную систему счисления;
- полученный результат дополнить слева незначащими нулями до k -разрядов.

Пример 38. Для хранения целого числа используется 1 байт. Каково внутреннее представление числа 83?

- 1) 01001011 2) 01100101
3) 01010011 4) 00101001

Решение. 1 байт содержит 8 бит, значит, во внутреннем представлении числа 8 разрядов. В двоичной системе счисления число 83 представимо в виде 1010011. В числе 7 разрядов, следовательно, нужно слева дописать один ноль 01010011.

Верный ответ: 3).

В случае отрицательного целого числа требуется:

- перевести модуль числа в двоичную систему счисления;
- получить обратный код этого числа заменой 0 на 1 и 1 на 0;
- к полученному числу прибавить 1.

Пример 39. Для хранения целого числа со знаком используется 1 байт. Сколько единиц содержит внутреннее представление числа (−35)?

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

Решение. 1 байт содержит 8 бит, значит, во внутреннем представлении числа 8 разрядов. Внутреннее представление числа 35 имеет вид 00100011. Обратный код имеет вид 11011100. После прибавления 1 получаем 11011101. Число единиц: 6.

Верный ответ: 2).

1.6.2. Кодирование текстовой информации

В текстовых документах широко используются не только русские, но и латинские буквы, цифры, математические знаки и другие специальные знаки, всего около 200 символов. Поэтому для кодировки всех указанных символов используется восьмиразрядная последовательность цифр 0 и 1. Таким образом, **текстовая информация** кодируется с использованием *кодовой таблицы*, где указаны все допустимые для использования символы и их коды.

Самая распространенная и универсальная компьютерная кодовая таблица ASCII (American Standard Code for Information Interchange) состоит из 16 строк и 16 столбцов, пронумерованных от 0 до F в 16-ричной системе счисления. Например, в столбце 4 и строке D таблицы расположена заглавная буква M латинского алфавита. Таким образом, при записи текста с такой буквой она будет храниться в памяти в виде кода $4D_{16}$ или 77_{10} . Обычно последние 8 столбцов таблицы кодов содержат буквы национальных алфавитов, графические знаки. В большом количестве разновидностей таблицы кодов ASCII первая половина таблицы является неизменной, а вторая — различается. Латинский алфавит упорядочен и непрерывен, то есть алфавитному порядку символов соответствует возрастающая последовательность кодов этих символов.

Обратите внимание на следующую задачу, представленную в демонстрационном варианте ЕГЭ 2010.

Пример 40. В таблице ниже представлена часть кодовой таблицы ASCII:

Символ	1	5	A	B	Q	a	b
Десятичный код	49	53	65	66	81	97	98
Шестнадцатеричный код	31	35	41	42	51	61	62

Каков шестнадцатеричный код символа «q»?

- 1) 71 2) 83 3) A1 4) B3

Решение.

Из таблицы видим, что разница в кодировке больших и малых букв составляет $97 - 65 = 32_{10}$. Поэтому десятичный код символа «q»: $81 + 32 = 113$. Переведем число 113_{10} в шестнадцатеричную систему счисления.

Верный ответ: 1).

1.6.3. Кодирование графической информации

Для представления *графической информации* в двоичной форме используется так называемый поточечный способ. На первом этапе вертикальными и горизонтальными линиями делят изображение. Чем больше при этом получилось квадратов, тем точнее будет передана информация о картинке. **Растровые изображения** представляют собой однослойную сетку точек, называемых пикселями (pixel, от англ. picture — element).

Код пикселя содержит информацию о его цвете. Для черно-белого изображения (без полутонов) пиксель может принимать только два значения: белый и черный (светится — не светится), а для его кодирования достаточно одного бита памяти: 1 — белый, 0 — черный.

Количество цветов K и количество битов N , отводимых в памяти для хранения каждого пикселя, связаны формулой:

$$2^N = K. \quad (8)$$

Например, если один пиксель изображения в памяти компьютера занимает 4 бита, то для создания этого изображения используется $K = 2^4 = 16$ цветов. Количество битов, необходимых для кодирования цвета одного пикселя, называется битовой глубиной цвета.

Пример 41. Для хранения растрового изображения размером 32×32 пикселя отвели 512 байтов памяти. Каково максимально возможное число цветов в палитре изображения?

- 1) 256 2) 2 3) 16 4) 4

Решение. Число точек изображения равно $32 \cdot 32 = 1024$. Мы знаем, что 512 байтов = $512 \cdot 8 = 4096$ бит. Найдем глубину цвета $4096 : 1024 = 4$. Число цветов равно $2^4 = 16$.

Верный ответ: 3).

Пример 42. Книга занимает объем 2 Мбайта. 256 страниц книги полностью заняты текстом. Каждая такая страница содержит ровно 1024 символа. 4 страницы полностью заполнены изображениями с разрешением 768 на 1024 точек. Сколько цветов в палитре изображений, если известно, что текстовые символы кодируются двухбайтной кодировкой?

Решение. Страницы с текстом занимают в памяти

$$256 \cdot 1024 \cdot 2 \text{ байта} = 2^8 \cdot 2^{10} \cdot 2 \text{ байта} = 2^{19} \text{ байта} = 0,5 \text{ Мбайта.}$$

На страницы с изображением остается $1,5$ Мбайта. Число точек изображения равно

$$768 \cdot 1024 = 3 \cdot 2^8 \cdot 2^{10} = 3 \cdot 2^{18}.$$

Умножаем число пикселей на число страниц и получаем $3 \cdot 2^{20}$ пикселей. Разделив

$$\frac{1,5 \text{ Мбайт}}{3 \cdot 2^{20} \text{ пиксель}} = \frac{1,5 \cdot 2^{20} \cdot 8 \text{ бит}}{3 \cdot 2^{20} \text{ пиксель}} = 4 \text{ бит/пиксель.}$$

Используя формулу (8), найдем глубину цвета $2^4 = 16$ цветов.

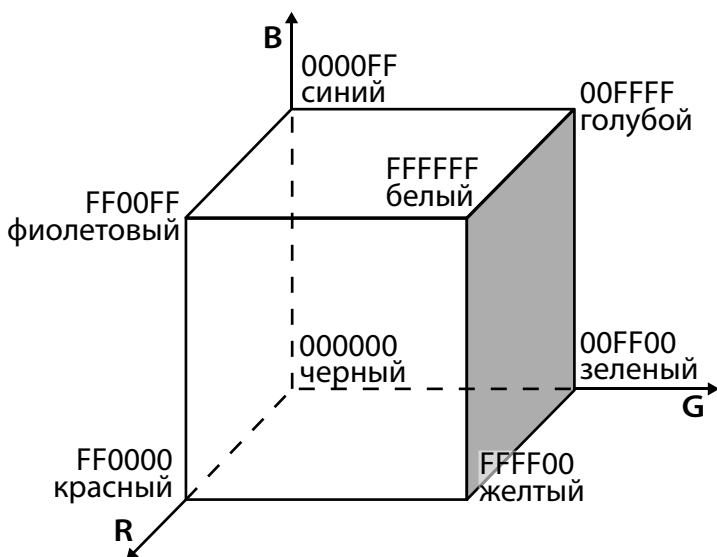
Верный ответ: 16.

Как известно из физики, любой цвет может быть представлен в виде суммы различной яркости зеленого, синего, красного цветов. Эта цветовая схема называется **RGB** (от англ. Red, Green, Blue — красный, зеленый, голубой). Каждый канал — R, G или B имеет свой отдельный параметр, указывающий на количество соответствующей компоненты в конечном цвете. При обозначении цветов в HTML-документах используется шестнадцатеричное обо-

значение 00, 33, 66, 99, FF, для графических редакторов используется обозначение 0, 51, 102, 153, 204, 255. При обозначении цветов в HTML-документах вначале ставят знак номера #. Таким образом, красный будет обозначаться как 255, 0, 0 в Photoshop'е и #FF0000 в HTML.

В HTML: #FF0000 — интенсивно красный цвет, #00FF00 — зеленый цвет, #0000FF — синий цвет. Отсутствие цветов (#000000) дает черный цвет, а самое интенсивное сочетание всех трех каналов (#FFFFFF) дает белый цвет.

Цветовым пространством RGB-модели является куб.



FF — наибольшая яркость цветовой компоненты, для получения различных оттенков одного и того же цвета изменяют яркость. Так : #FF00FF — интенсивно-фиолетовый цвет, : #990099 — темно-фиолетовый цвет.

Заметим, что если старший бит в коде (первая, третья или пятая цифра) находится в диапазоне от 0 до 3, то можно считать, что эта цветовая компонента отсутствует в цвете, то есть #0F0F0F — это черный цвет.

Также следует отметить, что равное, или почти равное, сочетание цветовых компонент обозначает серый цвет разной интенсивности.

Цветные изображения могут иметь различную битовую глубину цвета. 3 бита — по одному для каждого из трех базовых цветов — позволяют закодировать 8 цветов (так как $2^3 = 8$), а для 256 цветов нужно 8 бит памяти (так как $256 = 2^8$). Наиболее распространенными значениями глубины цвета являются 8, 16, 24 или 32 бита.

Пример 43. 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=«#00F000»>`?

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

Верный ответ: 2).

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Измерение количества информации

1. Считая, что каждый символ кодируется одним байтом, определите, чему равен информационный объем следующего высказывания *Алексея Толстого*:

Не ошибается тот, кто ничего не делает, хотя это и есть его основная ошибка.

- 1) 512 бит
- 2) 608 бит
- 3) 4 Кбайта
- 4) 123 байта

2. Считая, что каждый символ кодируется одним байтом, определите, чему равен информационный объем следующего высказывания *Рене Декарта*:

Я мыслю, следовательно, существую.

- 1) 28 бит
- 2) 272 бита
- 3) 32 Кбайта
- 4) 34 бита

3. Считая, что каждый символ кодируется 16 битами, оцените информационный объем следующей пушкинской фразы в кодировке Unicode:

Привычка свыше нам дана: Замена счастию она.

- 1) 44 бита
- 2) 704 бита
- 3) 44 байта
- 4) 704 байта

4. 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?

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

5. Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 8-битном коде, в 16-битную кодировку *Unicode*. При этом информационное сообщение увеличилось на 2048 байт. Каков был информационный объем сообщения до перекодировки?

- 1) 1024 байт
- 2) 2048 бит
- 3) 2 Кбайта
- 4) 2 Мбайта

6. Известно, что на каждой странице документа 128 строк, содержащих по 48 символов каждая. Сколько страниц в документе, если его информационный объем при условии, что каждый символ кодировался двухбайтовой кодировкой *Unicode*, составил 720 килобайт?

7. Сколько бит информации несет сообщение о том, что тетраэдр, у которого все грани окрашены в разные цвета, после подбрасывания упадет на синюю грань?

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

8. Два текста содержат одинаковое количество символов. Первый текст составлен в алфавите мощностью 32 768 символов, второй текст — в алфавите мощностью 8 символов. Во сколько раз количество информации в первом тексте больше второго? В ответе укажите число.

9. Мощность алфавита равна 64. Сколько Кбайт памяти потребуется, чтобы сохранить 128 страниц текста, содержащего в среднем 256 символов на каждой странице?

- 1) 8
- 2) 12
- 3) 24
- 4) 36

10. Для передачи секретного сообщения используется код, состоящий из десятичных цифр. При этом все цифры кодируются одним и тем же (минимально возможным) количеством бит. Определите информационный объем сообщения длиной в 150 символов.

- 1) 600 бит
- 2) 750 бит
- 3) 1200 бит
- 4) 60 байт

11. Дан текст, состоящий из 2000 символов. Известно, что символы берутся из таблицы размером 64×8 . Определить информационный объем текста в битах.

12. Для кодирования секретного сообщения используются 12 специальных значков-символов. При этом символы кодируются одним и тем же минимально возможным количеством бит. Чему равен информационный объем сообщения длиной в 256 символов?

- 1) 256 бит
- 2) 4000 бит
- 3) 56 байт
- 4) 128 байт

13. 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

14. Каждая клетка поля 8×8 кодируется минимально возможным и одинаковым количеством бит. Решение задачи о прохождении «конем» поля записывается последовательностью кодов посещенных клеток. Каков объем информации после 11 сделанных ходов? (Запись решения начинается с начальной позиции коня.)

- 1) 64 бит
- 2) 9 байт
- 3) 12 байт
- 4) 96 байт

15. В некоторой стране автомобильный номер состоит из 7 символов. В качестве символов используют 18 различных букв и десятичные цифры в любом порядке. Каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым целым количеством байтов, при этом используют посимвольное кодирование и

все символы кодируются одинаковым и минимально возможным количеством битов. Определите объем памяти, отводимый этой программой для записи 60 номеров.

- 1) 240 байт
- 2) 300 байт
- 3) 360 байт
- 4) 420 байт

16. Скорость передачи данных через ADSL-соединение равна 1024000 бит/с. Передача файла через данное соединение заняла 5 секунд. Определите размер файла в килобайтах.

17. 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 Кбит/с (впишите в бланк только число).

18. Скорость передачи данных через ADSL-соединение равна 256000 бит/с. Передача файла через это соединение заняла 2 минуты. Определите размер файла в килобайтах.

19. У Васи есть доступ к Интернету по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения им информации 2^{18} бит в секунду. У Пети нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^{15} бит в секунду. Петя договорился с Васей, что тот будет скачивать для него данные объемом 5 Мбайт по высокоскоростному каналу и ретранслировать их Пете по низкоскоростному каналу. Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 512 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Петей? В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

20. Три друга — Ваня, Коля и Сергей — совместно используют канал доступа в Интернет с пропускной способностью 96 Мбайт в секунду. Система балансировки нагрузки настроена таким образом, что если в данный момент времени канал использует только один человек, то скачивание файла происходит со скоростью, равной пропускной способности канала, а если канал используют несколько человек, пропускная способность канала поровну делится между пользователями. Ваня начал скачивать файл размером 3,5 Гбайт. Через 8 секунд Коля начал скачивать файл размером 2 Гбайт. Через 16 секунд после этого Сергей начал скачивать файл размером 512 Мбайт. Через какое время от начала скачивания Ваня полностью скачает свой файл? В ответе укажите целое число секунд.

Кодирование информации (комбинаторика)

1. Алфавит 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

2. Световое табло состоит из лампочек, каждая из которых может находиться в двух состояниях («включено» или «выключено»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 50 различных сигналов?

- 1) 5 2) 6 3) 25 4) 50

3. Сколько различных последовательностей длиной 7 символов можно составить из цифр 0 и 1?

- 1) 32 2) 64 3) 100 4) 128

4. Светодиод может находиться в трех состояниях. Какое минимальное количество светодиодов понадобится, чтобы собрать индикатор, способный воспроизводить 100 различных сообщений?

5. Некоторое сигнальное устройство за одну секунду передает один из трех сигналов. Сколько различных сообщений длиной в четыре секунды можно передать при помощи этого устройства?

6. Для передачи сигналов на флоте используются специальные сигнальные флаги, вывешиваемые в одну линию (последовательность важна). Какое количество различных сигналов может передать корабль при помощи четырех сигнальных флагов, если на корабле имеются флаги трех различных видов (флагов каждого вида неограниченное количество)?

7. Вася и Петя передают друг другу сообщения, используя синий, красный и зеленый фонарики. Это они делают, включая по одному фонарику на одинаковое короткое время в некоторой последовательности. Количество вспышек в одном сообщении — 3 или 4, между сообщениями — паузы. Сколько различных сообщений могут передавать мальчики?

8. Для кодирования 300 различных сообщений используются 5 последовательных цветовых вспышек. Вспышки одинаковой длительности, для каждой вспышки используется одна лампочка определенного цвета. Лампочки скольких цветов должны использоваться при передаче (укажите минимально возможное количество)?

9. Сколько существует четырехзначных чисел, составленных из разных четных цифр?

- 1) 96 2) 12 3) 500 4) 625

10. Сколько существует четырехзначных чисел, которые делятся на 5?

- 1) 900 2) 1000 3) 1800 4) 2000

11. В чемпионате по шахматам участвовало 40 спортсменов. Каждый с каждым сыграл по одной партии. Сколько всего партий было сыграно?

- 1) 780 2) 800 3) 1560 4) 1600

12. В вазе лежат яблоко, груша, персик и абрикос. Кате разрешили выбрать два каких-то фрукта. Сколько у Кати вариантов выбора?

- 1) 6 2) 12 3) 16 4) 24

13. У Паши есть 6 воздушных шариков разного цвета. Три из них он хочет подарить Маше. Сколькими способами он может это сделать?

- 1) 6 2) 12 3) 20 4) 60

14. Сколько существует четырехзначных чисел, которые читаются одинаково «слева направо» и «справа налево»?

- 1) 50 2) 90 3) 100 4) 120

15. Сколькими способами можно выложить в ряд красный, черный, синий и зеленый шарики?

16. Есть пять шариков: красный, зеленый, желтый, синий и золотой. Сколькими способами можно украсить ими пять елок, если на каждую требуется надеть ровно один шарик?

17. На пустую шашечную доску надо поместить две шашки разного цвета. Сколько различных положений могут они занимать на доске?

Представление числовой информации

1. Как записывается число 754_8 в шестнадцатеричной системе счисления?

- 1) 738_{16} 3) $1EC_{16}$
2) $1A4_{16}$ 4) $A56_{16}$

2. Как представлено число 25_{10} в двоичной системе счисления?

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

3. Дано: $a = 67_{16}$, $b = 151_8$. Какое из чисел c , записанных в двоичной системе, отвечает условию $a < c < b$?

- 1) 1101000
- 2) 1101010
- 3) 1101011
- 4) 1011000

4. 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

5. Вычислите сумму чисел x и y , при $x = A6_{16}$, $y = 75_8$. Результат представьте в двоичной системе счисления.

- 1) 11011011_2
- 2) 11110001_2
- 3) 11100011_2
- 4) 10010011_2

6. Вычислите сумму чисел x и y , если $x = 271_8$, $y = 11110100_2$. Результат представьте в шестнадцатеричной системе счисления.

- 1) 151_{16}
- 2) $1AD_{16}$
- 3) 412_{16}
- 4) $10B_{16}$

7. Вычислите разность $X - Y$ двоичных чисел, если $X = 1010100_2$, $Y = 1000010_2$. Результат представьте в двоичном виде.

- 1) 11010_2
- 2) 10100_2
- 3) 10010_2
- 4) 10101_2

8. Чему равна разность чисел $x = 67_8$ и $y = 1011101_2$?

- 1) 101010_2
- 2) 101000_2
- 3) 100110_2
- 4) 100101_2

9. Чему равна разность чисел $x = 101_{16}$ и $y = 1100101_2$?

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

10. В системе счисления с некоторым основанием число 27 записывается в виде 102. Укажите это основание.

11. В системе счисления с некоторым основанием число 121 записывается в виде 100. Укажите это основание.

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

13. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 39 оканчивается на 3.

14. Укажите наименьшее основание системы счисления, в которой запись числа 152 трехзначна.

15. Десятичное число, переведенное в восьмеричную и в шестеричную систему, в обоих случаях заканчивается на цифру 0. Какое минимальное натуральное число удовлетворяет этому условию?

16. Укажите, сколько всего раз встречается цифра 1 в записи чисел 12, 13, 14, ..., 31 в системе счисления с основанием 5.

17. Найдите основание системы счисления, в которой выполнено сложение: $144 + 24 = 201$.

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

В	К	А	Р	Д
000	11	01	001	10

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

- 1) 110100000100110011
- 2) 111010000010010011
- 3) 110100001001100111
- 4) 110110000100110010

19. 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	110	01	001	10

Определите, какой набор букв закодирован двоичной строкой 1100000100110.

- 1) Baade
- 2) bacde
- 3) bacdb
- 4) badde

20. Для кодирования букв А, Б, В, Г используются четырехразрядные последовательные двоичные числа от 1000 до 1011 соответственно. Если таким способом закодировать последовательность символов БГАВ и записать результат в восьмеричном коде, то получится:

- 1) 175423
- 2) 115612
- 3) 62577
- 4) 12376

Кодирование информации

1. На запись числа отводится 1 байт. Количество значащих нулей во внутреннем представлении десятичного числа 126 равно

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

2. Для хранения целого числа со знаком используется один байт. Сколько единиц содержит внутреннее представление числа (-128)?

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

3. В таблице ниже представлена часть кодовой таблицы ASCII:

Символ	1	5	A	B	U	a	b
Десятичный код	49	53	65	66	85	97	98
Шестнадцатеричный код	31	35	41	42	55	61	62

Каков шестнадцатеричный код символа «u»?

- 1) 8D 2) A5 3) 75 4) 73

4. В таблице ниже представлена часть кодовой таблицы ASCII:

Символ	1	5	A	B	A	b	s
Десятичный код	49	53	65	66	97	98	115
Шестнадцатеричный код	31	35	41	42	61	62	73

Каков шестнадцатеричный код символа «S»?

- 1) 72 2) 53 3) A2 4) 53

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

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

6. Цвет пикселя, формируемого принтером, определяется тремя составляющими: голубой, пурпурной и желтой. Под каждую составляющую одного пикселя отвели по четыре бита. В какое количество цветов можно раскрасить пиксель?

- 1) 12 2) 12^2 3) 2^{12} 4) 3^{12}

7. Цвет пикселя определяется тремя составляющими: зеленой, синей и красной. Под красную и синюю составляющие одного пикселя отвели по 5 бит. Сколько бит отвели под зеленую составляющую, если растровое изображение размером 8×8 пикселей занимает 128 байт памяти?

- 1) 5
2) 6
3) 8
4) 16

8. В процессе преобразования растрового графического файла количество цветов уменьшилось с 512 до 8. Во сколько раз уменьшился информационный объем файла?

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

9. Для кодирования цвета фона web-страницы 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=«#00FF00»>`?

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

10. Для кодирования цвета фона web-страницы 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=«#00FFF8»>?`

- 1) голубой
- 2) желтый
- 3) красный
- 4) фиолетовый

11. Для кодирования цвета фона web-страницы 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=«#EE22EE»>?`

- 1) голубой
- 2) желтый
- 3) красный
- 4) фиолетовый

12. Для кодирования цвета фона web-страницы 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=«#999900»>?`

- 1) голубой
- 2) желтый
- 3) красный
- 4) фиолетовый

13. Для кодирования цвета фона web-страницы 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=«#7F7F7F»>?`

- 1) белый
- 2) желтый
- 3) черный
- 4) серый

14. Для кодирования цвета фона web-страницы 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=«#00DDDD»>`?

- 1) голубой
- 2) желтый
- 3) черный
- 4) серый

15. Для кодирования цвета фона web-страницы 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=«#0A0A0A»>`?

- 1) голубой
- 2) черный
- 3) красный
- 4) серый

16. Скорость передачи данных через модемное соединение равна 12800 бит/с. Передача текстового файла через это соединение заняла 14 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке *Unicode*.

17. Средняя скорость передачи данных с помощью модема равна 36864 бит/с. Сколько секунд понадобится модему, чтобы передать 2 страницы текста в 8-битной кодировке КОИ8, если считать, что на каждой странице в среднем 2304 символа?

18. Сколько секунд потребуется модему, передающему сообщения со скоростью 14400 бит/с, чтобы передать цветное растровое изображение размером 640 на 480 пикселей, при условии, что цвет каждого пикселя кодируется 24 битами?

19. Графический файл, имеющий размер 1024×768 пикселей, кодированный без сжатия с использованием палитры из 65536 цветов, был передан через сетевое подключение с пропускной способностью канала 2 Кбайта в секунду. Во сколько раз уменьшится время его передачи по этому же каналу, если кодирование файла произвести с использованием палитры из 16 цветов?

Раздел 2

ОСНОВЫ ЛОГИКИ

2.1. ОСНОВНЫЕ ПОНЯТИЯ ЛОГИКИ

Алгебра логики — это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Логическое высказывание — это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Так, например, предложение «5 больше 3» следует считать высказыванием, так как оно истинное. Предложение «Москва — столица Франции» тоже высказывание, но оно ложное.

Разумеется, не всякое предложение является логическим высказыванием. Высказыванием не является, например, предложение «дождь со снегом», так как оно ничего не утверждает.

Алгебра логики рассматривает любое высказывание только с одной точки зрения — является ли оно истинным или ложным.

Употребляемые в обычной речи слова и словосочетания «не», «и», «или», «если... , то», «тогда и только тогда» и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются **логическими связками**.

Высказывания, образованные из других высказываний с помощью логических связок, называются **составными**. Высказывания, не являющиеся составными, называются **элементарными**. Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности элементарных высказываний.

Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через A обозначено высказывание « $x \leq 10$ », а через B — высказывание « $x \geq 0$ ». Тогда составное высказывание « $0 \leq x \leq 10$ » можно кратко записать как A и B . Здесь «и» — логическая связка, A , B — логические переменные, которые могут принимать только два значения — «истина» или «ложь», обозначаемые, соответственно, «1» и «0».

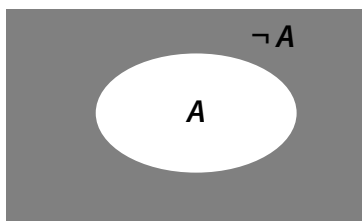
2.2. ЛОГИЧЕСКИЕ ОПЕРАЦИИ

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение.

2.2.1. Отрицание «НЕ»

Операция, выражаемая словом «не», называется отрицанием и обозначается чертой над высказыванием (или знаком \neg). Высказывание $\neg A$ (\bar{A}) истинно, когда A ложно, и ложно, когда A истинно. Например, если $A =$ « x — четное число», то $\neg A =$ « x — нечетное число».

Графически отрицание можно изобразить так: пусть высказывание A истинно для точек, принадлежащих овалу, тогда высказывание $\neg A$ истинно в выделенной области вне овала.



Запишем таблицу истинности для операции отрицания.

Таблица истинности — это табличное представление логической операции, в котором перечислены все возможные сочетания значений истинности входных операндов

вместе со значением истинности результата операции для каждого из этих сочетаний.

A	$\neg A$
0	1
1	0

2.2.2. Конъюнкция «И»

Операция, выражаемая связкой «и», называется **конъюнкцией** (лат. conjunctio — соединение) или **логическим умножением** и обозначается знаками \wedge или $\&$. Высказывание $A \wedge B$ *истинно тогда и только тогда, когда оба высказывания A и B истинны*. Например, высказывание «10 делится на 2 И 5 больше 3» истинно, а высказывание «10 делится на 2 И 5 не больше 3» ложно.

Графически конъюнкция высказываний A и B выглядит следующим образом: пусть высказывание A истинно для всех точек, принадлежащих овалу, а высказывание B — для точек прямоугольника, тогда высказывание $A \wedge B$ верно для выделенной области.

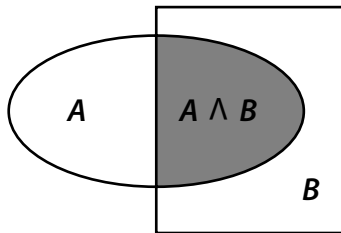


Таблица истинности операции Конъюнкция «И»

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Пример 1. Для какого символического выражения верно высказывание: \neg (Первая буква гласная) \wedge (Вторая буква согласная)?

- | | |
|----------|----------|
| 1) abcde | 3) babas |
| 2) bcade | 4) cabab |

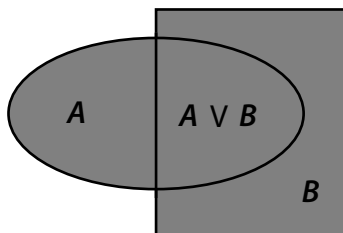
Решение. Так как конъюнкция истинна, то оба высказывания должны быть истинны. Первое высказывание построено с помощью операции отрицания, значит, выражение «Первая буква гласная» ложно, то есть первая буква не должна быть гласной, а вторая буква обязана быть согласной, таким требованиям удовлетворяет выражение «bcade».

Верный ответ: 2).

2.2.3. Дизъюнкция «ИЛИ»

Операция, выражаемая связкой «или» (в неисключающем смысле этого слова), называется **дизъюнкцией** (лат. disjunctio — разделение) или **логическим сложением** и обозначается знаком \vee . Высказывание $A \vee B$ **ложно тогда и только тогда, когда оба высказывания A и B ложны**. Так, высказывание «10 не делится на 2 ИЛИ 5 не больше 3» ложно, а высказывания «10 делится на 2 ИЛИ 5 больше 3», «10 делится на 2 ИЛИ 5 не больше 3», «10 не делится на 2 ИЛИ 5 больше 3» — истинны.

На рисунке дизъюнкция выглядит как объединение двух высказываний.



**Таблица истинности операции
Дизъюнкция «ИЛИ»**

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Пример 2. Для какого числа X истинно высказывание $(X \cdot (X - 16) \leq -64) \vee (X > 8)$?

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

Решение. Заметим, что высказывание $(X > 8)$ для любого из предложенных вариантов значений числа X ложно. Так как дизъюнкция истинна, то высказывание $X \cdot (X - 16) \leq -64$ должно быть истинно. Его можно преобразовать следующим образом $(X - 8)^2 \leq 0$. Квадрат разности не может быть отрицательным, а равен нулю только при условии $X = 8$.

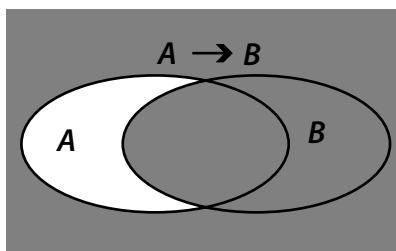
Верный ответ: 4).

2.2.4. Импликация «ЕСЛИ – ТО»

Операция, выражаемая связками «если... то», «из ... следует», «... влечет ...», называется импликацией (лат. *implicatio* — тесно связаны) и обозначается знаком \rightarrow . Высказывание $A \rightarrow B$ ложно тогда и только тогда, когда A истинно, а B ложно. Высказывание A называется посылкой импликации, а высказывание B — следствием.

В обычной речи связка «ЕСЛИ... ТО» описывает причинно-следственную связь между высказываниями. Но в логических операциях смысл высказываний не учитывается. Рассматривается только их истинность или ложность.

На рисунке видим результат импликации двух высказываний.



**Таблица истинности операции
Импликация «ЕСЛИ... ТО»**

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Пример 3. Для какого названия цвета ложно высказывание:

(Четвертая буква гласная) \rightarrow \neg (Вторая буква гласная)?

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

Решение. Импликация ложна в единственном случае, когда посылка (Четвертая буква гласная) истинна, а следствие \neg (Вторая буква гласная) ложно. Значит, истинны высказывания (Четвертая буква гласная) и (Вторая буква гласная). Из предложенных вариантов подходит только желтый.

Верный ответ: 2).

2.2.5. Равносильность «ТОГДА И ТОЛЬКО ТОГДА»

Операция, выражаемая связками «тогда и только тогда», «необходимо и достаточно», «... равносильно ...», называется **эквиваленцией** или двойной импликацией и обозначается знаком \leftrightarrow или \sim . Высказывание $A \leftrightarrow B$ истинно

тогда и только тогда, когда значения A и B совпадают. Высказывания A и B , образующие составное высказывание $A \leftrightarrow B$, могут быть совершенно не связаны по содержанию.

На рисунке выделена область, где истинно высказывание $A \leftrightarrow B$.

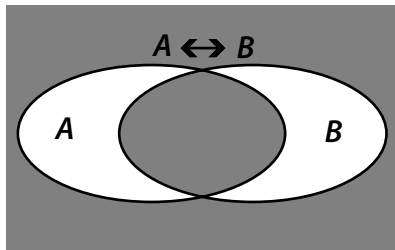


Таблица истинности операции Равносильности

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

2.2.6. Приоритет логических операций

Порядок выполнения логических операций задается круглыми скобками. Для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания («НЕ»), затем конъюнкция («И», логическое умножение), после конъюнкции — дизъюнкция («ИЛИ», логическое сложение) и в последнюю очередь — импликация и равносильность.

Пример 4. Для какого названия животного ложно высказывание:

$$(B \text{ слове } 4 \text{ гласные буквы}) \wedge \neg (\text{Пятая буква гласная}) \vee \vee (B \text{ слове } 5 \text{ согласных букв})?$$

- | | |
|-------------|-------------|
| 1) шиншилла | 3) антилопа |
| 2) кенгуру | 4) крокодил |

Решение. В этой задаче очень важно помнить о приоритете логических операций, поэтому выполним операцию отрицания над вторым высказыванием и поставим дополнительные скобки:

$$((\text{В слове 4 гласные буквы}) \wedge (\text{Пятая буква согласная})) \vee \vee (\text{В слове 5 согласных букв}).$$

Последнее логическое действие — дизъюнкция. Она ложна в том случае, если ложны оба высказывания. Так как неверно, что в слове 5 согласных букв, то из четырех вариантов остается только два.

- | | |
|-------------|-------------|
| 1) шиншилла | 3) антилопа |
| 2) кенгуру | 4) крокодил |

Третье слово содержит 4 гласные буквы, и, кроме того, пятая буква согласная, значит, высказывание *(В слове 4 гласные буквы) ∧ (Пятая буква согласная)* истинно, а должно быть ложно. То есть третий вариант можно вычеркнуть:

- | | |
|-------------|-------------|
| 1) шиншилла | 3) антилопа |
| 2) кенгуру | 4) крокодил |

Остался единственный вариант, в котором 3 гласные буквы, а значит, конъюнкция будет ложной.

Верный ответ: 2).

С помощью логических переменных и символов логических операций любое высказывание можно формализовать, то есть заменить логической формулой. **Логическая формула** — это формула, содержащая лишь логические величины и знаки логических операций. Результатом вычисления логической формулы является «истина» («1») или «ложь» («0»).

Некоторые формулы принимают значение «истина» при любых значениях истинности входящих в них переменных. Таковой будет, например, формула $A \vee \neg A$, соответствующая высказыванию «*Этот треугольник прямоугольный ИЛИ тупоугольный*». Эта формула истинна и тогда, когда треугольник прямоугольный, и тогда, когда треугольник не прямоугольный. Такие формулы называ-

ются **тождественно истинными** формулами или **тавтологиями**.

В качестве другого примера рассмотрим формулу $A \wedge \neg A$, которой соответствует, например, высказывание «*Катя самая высокая девочка в классе, и в классе есть девочки выше Кати*». Очевидно, что эта формула ложна, так как либо A , либо $\neg A$ обязательно ложно. Такие формулы называются **тождественно ложными** формулами или **противоречиями**.

Если две формулы A и B одновременно, то есть при одинаковых наборах значений входящих в них переменных, принимают одинаковые значения, то они называются **равносильными**.

Равносильность двух формул алгебры логики обозначается символом « $=$ » или символом « \equiv ». Замена формулы другой, ей равносильной, называется равносильным преобразованием данной формулы.

Замечание 1. Импликацию можно заменить дизъюнкцией: выражения $A \rightarrow B$ и $(\neg A) \vee B$ равносильны, т.е. $A \rightarrow B = (\neg A) \vee B$.

2.3. ОСНОВНЫЕ ЗАКОНЫ АЛГЕБРЫ ЛОГИКИ

В алгебре логики выполняются следующие основные законы, позволяющие производить тождественные преобразования логических выражений:

Закон	Для дизъюнкции	Для конъюнкции
Переместительный	$A \vee B = B \vee A$	$A \wedge B = B \wedge A$
Сочетательный	$A \vee (B \vee C) = (A \vee B) \vee C$	$A \wedge (B \wedge C) = (A \wedge B) \wedge C$
Распределительный	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
Правила де Моргана	$\neg(A \vee B) = \neg A \wedge \neg B$	$\neg(A \wedge B) = \neg A \vee \neg B$
Идемпотенции	$A \vee A = A$	$A \wedge A = A$
Поглощения	$A \vee (A \wedge B) = A$	$A \wedge (A \vee B) = A$

Закон	Для дизъюнкции	Для конъюнкции
Склеивания	$(A \wedge B) \vee \vee (-A \wedge B) = B$	$(A \vee B) \wedge \wedge (-A \vee B) = B$
Операция переменной с ее инверсией	$A \vee \neg A = 1$	$A \wedge \neg A = 0$
Операция с константами	$A \vee 0 = A;$ $A \vee 1 = 1$	$A \wedge 0 = 0;$ $A \wedge 1 = A$
Двойного отрицания	$\neg \neg A = A$	

Замечание 2. Из правила де Моргана и равносильности $A \rightarrow B = (\neg A) \vee B$ получаем $\neg(A \rightarrow B) = A \wedge (\neg B)$.

Равносильные преобразования логических формул имеют то же назначение, что и преобразования формул в обычной алгебре. Они служат для упрощения формул или приведения их к определенному виду путем использования основных законов алгебры логики.

Под упрощением формулы, не содержащей операций импликации и эквиваленции, понимают равносильное преобразование, приводящее к формуле, которая

- либо содержит по сравнению с исходной меньшее число операций конъюнкции и дизъюнкции и не содержит отрицаний неэлементарных формул,
- либо содержит меньшее число вхождений переменных.

Некоторые преобразования логических формул похожи на преобразования формул в обычной алгебре (вынесение общего множителя за скобки, использование переместительного и сочетательного законов и т.п.), тогда как другие преобразования основаны на свойствах, которыми не обладают операции обычной алгебры (использование распределительного закона для конъюнкции, законов поглощения, склеивания, де Моргана и др.).

Покажем на примерах некоторые приемы и способы, применяемые при упрощении логических формул:

$$\begin{aligned} 1) \quad & \neg(A \vee B) \wedge (A \wedge \neg B) = \neg A \wedge \neg B \wedge (A \wedge \neg B) = \\ & = \neg A \wedge A \wedge \neg B \wedge \neg B = 0 \wedge \neg B \wedge \neg B = 0 \wedge \neg B = 0 \end{aligned}$$

(законы алгебры логики применяются в следующей последовательности: правило де Моргана, сочетательный закон, правило операций переменной с ее инверсией и правило операций с константами);

$$\begin{aligned} 2) \quad & \neg A \wedge B \vee \neg(A \vee B) \vee A = \neg A \wedge B \vee \neg A \wedge \neg B \vee A = \\ & = \neg A \wedge (B \vee \neg B) \vee A = \neg A \vee A = 1 \end{aligned}$$

(применяется правило де Моргана, выносится за скобки общий множитель, используется правило операций переменной с ее инверсией).

Пример 5. Какое логическое выражение равносильно выражению $\neg(\neg A \wedge B \wedge \neg C)$?

$$\begin{array}{ll} 1) A \wedge B \wedge \neg C & 3) \neg A \wedge B \wedge \neg C \\ 2) A \vee \neg B \vee C & 4) A \vee \neg B \vee \neg C \end{array}$$

Решение. По правилу де Моргана выполним преобразование $\neg(\neg A \wedge B \wedge \neg C) = \neg A \vee \neg B \vee \neg \neg C$. Пользуясь правилом двойного отрицания, в итоге получаем:

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

Верный ответ: 2).

Пример 6. Какое логическое выражение эквивалентно выражению $\neg(A \vee \neg B) \wedge \neg C$?

$$\begin{array}{ll} 1) A \vee B \wedge C & 3) \neg(A \vee C) \vee B \\ 2) \neg(A \wedge B) \wedge C & 4) \neg(A \vee C) \wedge B \end{array}$$

Решение. Применим правило де Моргана и правило двойного отрицания

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

Однако такой ответ отсутствует в приведенных вариантах. Продолжим преобразования, используя переместительный и сочетательный законы:

$$\begin{aligned} (\neg A \wedge B) \wedge \neg C &= \wedge \neg C \wedge (\neg A \wedge B) = (\neg C \wedge \neg A) \wedge B = \\ &= (\neg A \wedge \neg C) \wedge B. \end{aligned}$$

Снова воспользуемся правилом де Моргана

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

Верный ответ: 4).

Пример 7. Какое логическое выражение эквивалентно выражению $\neg (A \vee \neg B \wedge C)$?

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

Решение. При выполнении этого задания очень легко ошибиться, если не учесть приоритет логических операций. Поставим дополнительные скобки, затем дважды применим закон де Моргана и правило двойного отрицания

$$\begin{aligned} \neg (A \vee (\neg B \wedge C)) &= \neg A \wedge \neg (\neg B \wedge C) = \\ &= \neg A \wedge (\neg \neg B \vee \neg C) = \neg A \wedge (B \vee \neg C). \end{aligned}$$

Продолжим преобразования, используя распределительный закон:

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

Верный ответ: 4).

Пример 8. Какое из приведенных имен удовлетворяет логическому условию \neg (Последняя буква гласная \rightarrow Вторая буква гласная) \wedge Первая буква согласная?

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

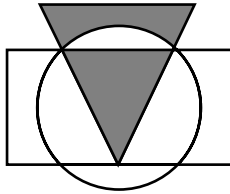
Решение. Преобразуем логическую формулу, используя замечание 2:

\neg (Последняя буква гласная \rightarrow Вторая буква гласная) \wedge Первая буква согласная = Последняя буква гласная \wedge \neg (Вторая буква гласная) \wedge Первая буква согласная = Последняя буква гласная \wedge Вторая буква согласная \wedge Первая буква согласная. Этим трем условиям удовлетворяет имя КСЕНИЯ.

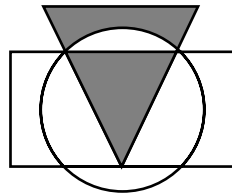
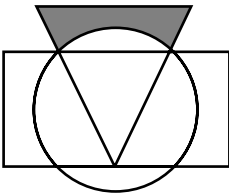
Верный ответ: 1).

Пример 9. Высказывания A , B и C истинны для точек, принадлежащих соответственно кругу, треугольнику и прямоугольнику соответственно. Для всех точек выделенной на рисунке области истинно высказывание

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



Решение. Выделенная область находится внутри треугольника, значит, высказывание B должно быть без отрицания (вычеркиваем третий вариант ответа). Кроме того, выделенная область не выходит за пределы треугольника, следовательно, не подойдет второй вариант ответа, который помимо треугольника включает еще и внешнюю часть круга и прямоугольника. Первый вариант ответа состоит из точек треугольника без точек, не вошедших одновременно в круг и прямоугольник (см. рисунок справа). Остается четвертый вариант. К области внутри прямоугольника добавляется область снаружи круга, а затем среди этих точек оставляют только те, что находятся внутри треугольника (см. рисунки ниже).



Именно эту область мы и искали.

Верный ответ: 4).

Пример 10. Для какого из указанных значений X истинно высказывание $\neg((X > 2) \rightarrow (X > 3))$?

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

Решение. Высказывание $\neg((X > 2) \rightarrow (X > 3))$ истинно, значит, по определению, высказывание $(X > 2) \rightarrow (X > 3)$ ложно. Импликация ложна, только если посылка истинна, а следствие ложно. Получаем систему

$$\begin{cases} X > 2, \\ X \leq 3. \end{cases} \quad 2 < X \leq 3, \quad \text{значит, } X = 3.$$

Верный ответ: 1).

Заметим, что задание 11 можно решить с помощью таблицы истинности, для этого проверим истинность следующих выражений

	$X > 2$	$X > 3$	$(X > 2) \rightarrow (X > 3)$	$\neg((X > 2) \rightarrow (X > 3))$
$X = 1$	0 (ложь)	0	1	0
$X = 2$	0	0	1	0
$X = 3$	1 (истина)	0	0	1
$X = 4$	1	1	1	0

Видим, что результат истинен только при $X = 3$.

Пример 11. Упростите логическое выражение

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

и укажите его результат. Результат упрощения может содержать только операции инверсии, конъюнкции и дизъюнкции.

Решение. Избавимся от импликации

$$\begin{aligned} \neg(A \wedge B \vee (A \rightarrow B)) \rightarrow B &= \neg(A \wedge B \vee (A \rightarrow B)) \vee B = \\ &= (A \wedge B \vee (A \rightarrow B)) \vee B = (A \wedge B \vee (\neg A \vee B)) \vee B. \end{aligned}$$

Применим переместительный закон

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

По закону поглощения получаем

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

Далее $(B \vee \neg A) \vee B = B \vee B \vee \neg A = B \vee \neg A$.

Верный ответ: $B \vee \neg A$.

Пример 12. Какая из данных логических формул является тождественно истинной?

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

Решение. Самый простой способ решить данную задачу — это построить таблицу истинности, которая выражает соответствие между всевозможными наборами значений переменных и значениями формул. Для формулы, содержащей две переменные, таких наборов значений переменных всего $2^2 = 4$: $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$.

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

$A \wedge (A \rightarrow B)$	$A \rightarrow (A \wedge B)$	$(A \rightarrow B) \vee A$	$(A \vee B) \rightarrow A$
0	0	1	1
		1	0
		1	
		1	

Заметим, что нет смысла заполнять таблицу полностью. Как только обнаружено несоответствие с условием задачи, можно переходить к следующей формуле.

Верный ответ: 3).

Строить таблицу истинности необходимо и при решении следующей задачи.

Пример 13. Дан фрагмент таблицы истинности выражения F :

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

Каким выражением может быть F ?

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

Решение. Построим таблицу истинности для всех предложенных в ответе выражений:

X	Y	Z	$X \wedge Y \wedge Z$	$\neg X \vee \neg Y \vee Z$	$X \vee Y \vee Z$	$X \wedge Y \wedge \neg Z$	F
1	1	1	1	1	1	0	1
1	1	0	0	0	1	1	1
1	0	1	0	1	1	0	1

Для вычисления значений логических выражений в столбцах 4–7 необходимо применить таблицы истинности логических выражений. Видим, что совпали значения логических выражений в столбцах $X \vee Y \vee Z$ и F .

Верный ответ: 3).

Замечание 3. Следует отметить, что при составлении таблицы истинности нет необходимости заполнять все ячейки. Как только обнаружилось несовпадение значений со столбцом F , можно переходить к заполнению следующего столбца:

X	Y	Z	$X \wedge Y \wedge Z$	$\neg X \vee \neg Y \vee Z$	$X \vee Y \vee Z$	$X \wedge Y \wedge \neg Z$	F
1	1	1	1	1	1	0	1
1	1	0	0	0	1		1
1	0	1			1		1

Пример 14. Каково наибольшее целое число X , при котором истинно высказывание $(X \cdot X - 1 > 100) \rightarrow (X \cdot (X - 1) < 100)$?

Решение. С помощью замечания 1 можем исключить импликацию

$$(X^2 > 101) \rightarrow (X^2 - X < 100) = \neg (X^2 > 101) \vee \\ \vee (X^2 - X < 100) = (X^2 \leq 101) \vee (X^2 - X - 100 < 0).$$

Так как дизъюнкция истинна, то получаем совокупность:

$$\left[\begin{array}{l} X^2 \leq 101, \\ X^2 - X - 100 < 0, \end{array} \right. \Rightarrow \left[\begin{array}{l} -\sqrt{101} \leq X \leq \sqrt{101}, \\ \frac{1 - \sqrt{401}}{2} \leq X \leq \frac{1 + \sqrt{401}}{2}, \end{array} \right. \Rightarrow \\ \Rightarrow \left[\begin{array}{l} -10 \leq X \leq 10, \\ -9,5 \leq X \leq 10,5, \end{array} \right. \Rightarrow -10 \leq X \leq 10,5.$$

Таким образом, наибольшее целое число X , удовлетворяющее этой совокупности, $X = 10$.

Верный ответ: 10.

Пример 15. Укажите значения переменных K, L, M, N , при которых логическое выражение $(\neg(M \vee L) \wedge K) \rightarrow ((\neg K \wedge \neg M) \vee N)$ ложно. Ответ запишите в виде строки из четырех символов: значение переменных K, L, M и N (в указанном порядке). Так, например, строка 0101 соответствует тому, что $K = 0, L = 1, M = 0$ и $N = 1$.

Решение. Выражение $(\neg(M \vee L) \wedge K) \rightarrow ((\neg K \wedge \neg M) \vee N) = 0$ тогда и только тогда, когда
$$\left\{ \begin{array}{l} \neg(M \vee L) \wedge K = 1, \\ (\neg K \wedge \neg M) \vee N = 0. \end{array} \right.$$

Конъюнкция истинна только в одном случае, если

$$\left\{ \begin{array}{l} \neg(M \vee L) = 1, \\ K = 1. \end{array} \right.$$

В единственном случае ложна и дизъюнкция

$$\left\{ \begin{array}{l} \neg K \wedge \neg M = 0, \\ N = 0. \end{array} \right.$$

В результате получаем систему

$$\left\{ \begin{array}{l} K = 1, \\ N = 0, \\ M \vee L = 0, \\ \neg K \wedge \neg M = 0. \end{array} \right. \Rightarrow \left\{ \begin{array}{l} K = 1, \\ N = 0, \\ L = 0, \\ M = 0. \end{array} \right.$$

Верный ответ: 1000.

Пример 16. 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.

Решение. Дизъюнкция истинна только в следующих случаях

$$\left\{ \begin{array}{l} K \wedge L \wedge M = 1, \\ \neg L \wedge \neg M \wedge N = 1, \end{array} \right.$$

или

$$\left\{ \begin{array}{l} K \wedge L \wedge M = 1, \\ \neg L \wedge \neg M \wedge N = 0, \end{array} \right.$$

или

$$\left\{ \begin{array}{l} K \wedge L \wedge M = 0, \\ \neg L \wedge \neg M \wedge N = 1. \end{array} \right.$$

В первом случае получается

$$\left\{ \begin{array}{l} K = 1, \\ L = 1, \\ M = 1, \\ \neg L = 1, \\ \neg M = 1, \\ N = 1, \end{array} \right. \quad \text{или} \quad \left\{ \begin{array}{l} K = 1, \\ L = 1, \\ M = 1, \\ L = 0, \\ M = 0, \\ N = 1. \end{array} \right.$$

Видим, что система не совместна.

Во втором случае получаем

$$\left\{ \begin{array}{l} K = 1, \\ L = 1, \\ M = 1, \\ \neg L \wedge \neg M \wedge N = 0, \end{array} \right. \quad \text{или} \quad \left\{ \begin{array}{l} K = 1, \\ L = 1, \\ M = 1, \\ 0 \wedge 0 \wedge N = 0. \end{array} \right.$$

Последняя строка верна при любом значении переменной N . Таким образом, система имеет два решения.

В третьем случае

$$\left\{ \begin{array}{l} K \wedge L \wedge M = 0, \\ \neg L = 1, \\ \neg M = 1, \\ N = 1, \end{array} \right. \quad \text{или} \quad \left\{ \begin{array}{l} K \wedge 0 \wedge 0 = 0, \\ \neg L = 1, \\ \neg M = 1, \\ N = 1, \end{array} \right.$$

уравнение $K \wedge 0 \wedge 0 = 0$ верно при любом значении K , то есть еще два решения системы.

Верный ответ: 4.

Пример 17. Сколько различных решений имеет уравнение

$\neg M \wedge K \wedge \neg N \wedge \neg J \wedge (L \vee \neg L) = 0$, где J, K, L, M, N — логические переменные? В ответе не нужно перечислять все различные наборы значений J, K, L, M и N , при которых выполнено данное равенство. В качестве ответа вам нужно указать только количество таких наборов.

Решение. Заметим, что выражение $L \vee \neg L$ истинно при любом значении переменной L , тогда выражение $\neg M \wedge K \wedge \neg N \wedge \neg J$ ложно. Конъюнкция истинна только в одном случае ($\neg M = 1, K = 1, \neg N = 1, \neg J = 1$), в остальных случаях выражение ложно. Количество этих случаев $2^4 - 1 = 15$. Таким образом, количество решений уравнения $15 \cdot 2 = 30$. Умножение на 2 происходит потому, что уравнение верно для любого значения L .

Верный ответ: 30.

2.4. РЕШЕНИЕ ЛОГИЧЕСКИХ ЗАДАЧ

Разнообразие логических задач очень велико. Способов их решения тоже немало. Но наибольшее распространение получили следующие три способа решения логических задач:

- 1) средствами алгебры логики;
- 2) табличный;
- 3) с помощью рассуждений.

2.4.1. Решение логических задач средствами алгебры логики

Обычно используется следующая схема решения:

- 1) изучается условие задачи;
- 2) вводится система обозначений для логических высказываний;
- 3) конструируется логическая формула, описывающая логические связи между всеми высказываниями условия задачи;
- 4) определяются значения истинности этой логической формулы;
- 5) из полученных значений истинности формулы определяются значения истинности введенных логических высказываний, на основании которых делается заключение о решении.

Пример 18. Трое друзей, болельщиков автогонок «Формула-1», спорили о результатах предстоящего этапа гонок.

— Вот увидишь, Шумахер не придет первым, — сказал Джон. — Первым будет Хилл.

— Да нет же, победителем будет, как всегда, Шумахер, — воскликнул Ник. — А об Алезе и говорить нечего, ему не быть первым.

Питер, к которому обратился Ник, возмущен:

— Хиллу не видать первого места, а вот Алезе пилотирует самую мощную машину.

По завершении этапа гонок оказалось, что каждое из двух предположений двоих друзей подтвердилось, а оба предположения третьего из друзей оказались неверны. Кто выиграл этап гонки?

Решение. Введем обозначения для логических высказываний:

$Ш$ — победит Шумахер; X — победит Хилл; A — победит Алези.

Реплика Ника «Алези пилотирует самую мощную машину» не содержит никакого утверждения о месте, которое займет этот гонщик, поэтому в дальнейших рассуждениях не учитывается.

Зафиксируем высказывания каждого из друзей:

Джон: $\neg Ш \wedge X$, Ник: $Ш \wedge \neg A$, Питер: $\neg X$.

Так как двое оказались правы, а один из болельщиков ошибся, то получаем следующие три набора выражений:

$(\neg Ш \wedge X) \wedge (Ш \wedge \neg A) \wedge \neg \neg X$ (ошибался третий болельщик),

$(\neg Ш \wedge X) \wedge \neg(Ш \wedge \neg A) \wedge \neg X$ (ошибался второй болельщик),

$\neg(\neg Ш \wedge X) \wedge (Ш \wedge \neg A) \wedge \neg X$ (ошибался первый болельщик).

Одно из трех выражений будет истинным, а значит, истинной будет дизъюнкция

$$\begin{aligned} & ((\neg Ш \wedge X) \wedge (Ш \wedge \neg A) \wedge \neg \neg X) \vee \\ & \vee ((\neg Ш \wedge X) \wedge \neg(Ш \wedge \neg A) \wedge \neg X) \vee \\ & \vee (\neg(\neg Ш \wedge X) \wedge (Ш \wedge \neg A) \wedge \neg X). \end{aligned}$$

Упростим это высказывание

$$\begin{aligned} & ((\neg Ш \wedge X) \wedge (Ш \wedge \neg A) \wedge \neg \neg X) \vee \\ & \vee ((\neg Ш \wedge X) \wedge \neg(Ш \wedge \neg A) \wedge \neg X) \vee \\ & \vee (\neg(\neg Ш \wedge X) \wedge (Ш \wedge \neg A) \wedge \neg X) = \\ & = (\neg Ш \wedge X \wedge Ш \wedge \neg A \wedge X) \vee \\ & \vee (\neg Ш \wedge X \wedge (\neg Ш \vee A) \wedge \neg X) \vee ((Ш \vee \neg X) \wedge Ш \wedge \neg A \wedge \neg X) = \\ & = (\neg Ш \wedge Ш \wedge X \wedge \neg A) \vee (\neg Ш \wedge X \wedge \neg X \wedge (\neg Ш \vee A)) \vee \\ & \vee ((Ш \wedge Ш \wedge \neg A \wedge \neg X) \vee (\neg X \wedge Ш \wedge \neg A \wedge \neg X)) = \end{aligned}$$

$$\begin{aligned}
 &= (0) \vee (0) \left((Ш \wedge Ш \wedge \neg A \wedge \neg X) \vee (\neg X \wedge Ш \wedge \neg A \wedge \neg X) \right) = \\
 &= \left((Ш \wedge \neg A \wedge \neg X) \vee (\neg X \wedge Ш \wedge \neg A) \right) = 1.
 \end{aligned}$$

Высказывание $\left((Ш \wedge \neg A \wedge \neg X) \vee (\neg X \wedge Ш \wedge \neg A) \right)$ истинно только при $Ш = 1, A = 0, X = 0$.

Другим способом решения этой задачи является составление таблицы истинности. Так как по условию задачи победит только один из трех игроков, то достаточно просмотреть три следующих набора данных. Для каждого набора данных определим истинность высказывания болельщиков.

$Ш$	X	A	$Ш \wedge \bar{X}$	$Ш \wedge \bar{A}$	\bar{A}
0	0	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	1

Ситуация, когда истинны два из трех высказываний, верна для последнего набора данных.

Ответ. Победителем этапа гонок стал Шумахер.

2.4.2. Решение логических задач табличным способом

При использовании этого способа условия, которые содержит задача, и результаты рассуждений фиксируются с помощью специально составленных таблиц.

Пример 19. На одной улице стоят в ряд 4 дома, в которых живут 4 человека: Алексей, Егор, Виктор и Михаил. Известно, что каждый из них владеет ровно одной из следующих профессий: Токарь, Столяр, Хирург и Окулист, но неизвестно, кто какой, и неизвестно, кто в каком доме живет. Однако известно, что:

- 1) Токарь живет через дом от Столяра
- 2) Хирург живет левее Токаря
- 3) Окулист живет правее Токаря
- 4) Хирург живет не рядом со Столяром
- 5) Михаил не Токарь

6) Алексей живет рядом с Окулистом

7) Егор живет справа от Токаря

8) Виктор живет рядом с Хирургом

Выясните, кто какой профессии и кто где живет, и дайте ответ в виде заглавных букв имени людей, в порядке слева направо. Например, если бы в домах жили (слева направо) Константин, Николай, Роман и Олег, ответ был бы: КНРО.

Решение. Договоримся, что при решении задачи будем использовать первую (заглавную) букву имени и профессии.

Из п. 2) следует, что дома расположены следующим образом **Х Т** (при этом не учитывается тот факт, что между домами Хирурга и Токаря могут располагаться другие дома).

Из п. 3) следует, что **Х Т О**.

Из условия п. 1) возможны два варианта **С Х Т О** или **Х Т О С**.

Условие п. 4) оставляет только один вариант: **Х Т О С**.

Для того чтобы определить, кто какой профессии, построим таблицу:

Из п. 8) следует, что Виктор — Токарь, так как рядом с Хирургом живет только Токарь.

	Х	Т	О	С
А		–		
Е		–		
В	–	+	–	–
М		–		

Из п. 6) получаем следующую информацию: Алексей не Окулист и не Хирург, так как Хирург не живет рядом с Окулистом.

	Х	Т	О	С
А	–	–	–	
Е		–		
В	–	+	–	–
М		–		

Получается, что Алексей — Столяр.

	Х	Т	О	С
А	–	–	–	+
Е		–		–
В	–	+	–	–
М		–		–

Из п. 7) следует, что Егор не Хирург.

	Х	Т	О	С
А	–	–	–	+
Е	–	–		–
В	–	+	–	–
М		–		–

Тогда однозначно получается, что Егор — Окулист, а Михаил — Хирург.

	Х	Т	О	С
А	–	–	–	+
Е	–	–	+	–
В	–	+	–	–
М	+	–	–	–

Верный ответ: МВЕА.

Пример 20. Три школьника, Миша (М), Коля (К) и Сергей (С), остававшиеся 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:

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

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

Миша: «Я не бил окно, и Коля тоже...»

Стало известно, что один из ребят сказал чистую правду, второй в одной части заявления соврал, а другое его высказывание истинно, а третий оба факта исказил. Зная это, директор смог докопаться до истины. Кто разбил стекло в классе? В ответе запишите только первую букву имени.

Решение. Предположим, что **окно разбил Коля**. Построим таблицу на основе высказываний мальчиков.

	Сказал чистую правду (+)	Сказал полуправду-полуложь (\pm)	Оба факта искажил (-)
Коля	–		–
Сергей	–		–
Миша			

Получили противоречие, так как оба мальчика исказили один из двух фактов.

Предположим, что **окно разбил Сергей**, тогда получаем таблицу.

	Сказал чистую правду (+)	Сказал полуправду-полуложь (\pm)	Оба факта искажил (-)
Коля		–	–
Сергей	–	–	
Миша		–	–

Получили противоречие, так как два мальчика сказали чистую правду.

Осталось проверить последнее предположение, что **окно разбил Миша**. В этом случае получаем таблицу.

	Сказал чистую правду (+)	Сказал полуправду-полуложь (\pm)	Оба факта искажил (-)
Миша	–		–
Коля	–	–	
Сергей		–	–

Здесь противоречий нет.

Верный ответ: М.

2.4.3. Решение логических задач с помощью рассуждений

Этим способом обычно решают несложные логические задачи.

Пример 21. Три молодые мамы Анна, Ирина и Ольга, гуляя в парке со своими малышами, встретили свою чет-

вертую подругу. На вопрос, как зовут малышей, желая подшутить над подружкой, они ответили:

Анна: Моего малыша зовут Денис, а Кирилл — сын Ирины.

Ирина: Моего сыночка зовут Максим, а Кирилл — сын Анны.

Ольга: Мой мальчик — Кирилл, а сына Анны зовут Максим.

Каждая из них один раз сказала правду и один раз солгала. Как зовут мальчиков Анны, Ирины и Ольги? В ответе перечислите подряд без пробелов буквы, соответствующие именам мальчиков в указанном порядке имен их мам, например КМД.

Решение. Обозначим фразу «у мамы по имени А сына зовут Б» как АБ, где буквы А и Б соответствуют начальным буквам имени.

Зафиксируем высказывания каждой мамы:

АД и ИК; ИМ и АК; ОК и АМ.

Предположим, что истинно ИК, то есть сына Ирины зовут Кириллом, тогда АД и ИК; из второго высказывания получаем, что ложны оба высказывания ИМ и АК, а это противоречит условию задачи.

Допустим, что истинно АД, то есть сына Анны зовут Денисом: АД и ИК, тогда ИМ и АК, ОК и АМ.

Верный ответ: ДМК.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Основы логики

1. Для какого из значений числа Z высказывание $((Z > 2) \vee (Z > 4)) \rightarrow (Z > 3)$ будет ложным?

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

2. Для какого из указанных значений числа X истинно высказывание

$(X < 5) \rightarrow (X < 3) \wedge ((X < 2) \rightarrow (X < 1))$

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

3. Какое из приведенных имен удовлетворяет логическому условию:

\neg (Вторая буква согласная \rightarrow Последняя буква гласная) \wedge Первая буква гласная?

- 1) ИРИНА
- 2) АРТЕМ
- 3) СТЕПАН
- 4) МАРИЯ

4. Для какого имени истинно высказывание:

(Первая буква согласная \rightarrow Вторая буква согласная) \wedge Последняя буква гласная?

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

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

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

6. 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) $A \wedge \neg B \wedge C$

7. Какое логическое выражение эквивалентно выражению $\neg A \wedge \neg(\neg B \vee \neg C) \vee D$?

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

8. Какое логическое выражение эквивалентно выражению $\neg(\neg A \vee \neg B \vee C)$?

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

9. 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) $\neg X \wedge \neg Y \wedge \neg Z$ 3) $X \vee Y \vee Z$
 2) $X \wedge Y \wedge Z$ 4) $\neg X \vee \neg Y \vee \neg Z$

10. Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X , Y , Z . Дан фрагмент таблицы истинности выражения F (см. таблицу внизу). Какое выражение соответствует F ?

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

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

11. Дано логическое выражение

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

Укажите значения переменных K , L , M , N , при которых логическое выражение **ложно**. Ответ запишите в виде строки из четырех символов: значения переменных K , L , M и N (в указанном порядке). Так, например, строка 0101 соответствует тому, что $K = 0$, $L = 1$, $M = 0$ и $N = 1$.

12. Сколько различных решений имеет уравнение $((K \vee L) \rightarrow (L \wedge M \wedge N)) = 0$, где K , L , M , N — логические переменные? В ответе не нужно перечислять все различные наборы значений K , L , M и N , при которых выполнено данное равенство. В качестве ответа вам нужно указать количество таких наборов.

13. Укажите значения логических переменных K , L , M , N , при которых логическое выражение

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

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

14. Упростите логическое выражение

$$\neg C \wedge D \vee \neg (C \vee D) \vee C$$

и укажите его результат. Результат упрощения может содержать только операции инверсии, конъюнкции и дизъюнкции.

15. Составьте таблицу истинности для логической функции $X = \neg(A \rightarrow B) \wedge (B \leftrightarrow \neg(C \rightarrow A))$, в которой столбец значений аргумента A представляет собой двоичную запись числа 216, столбец значений аргумента B — числа 30, столбец значений аргумента C — числа 170. Число в столбце записывается сверху вниз от старшего разряда к младшему. Переведите полученную двоичную запись значений функции X в десятичную систему счисления.

16. A , B и C — целые числа, для которых истинно высказывание:

$$(C < A \vee C < B) \wedge \neg(C + 1 < A) \wedge \neg(C + 1 < B).$$

Чему равно C , если $A = 45$ и $B = 18$?

17. Каково наибольшее целое положительное число X , при котором ложно высказывание:

$$(8 \cdot X - 6 < 75) \rightarrow (X \cdot (X - 1) > 65)?$$

18. На одной улице стоят в ряд 4 дома, в которых живут 4 человека: Алексей, Егор, Виктор и Михаил. Известно, что каждый из них владеет ровно одной из следующих профессий: Токарь, Столяр, Хирург и Окулист, но неизвестно, кто какой, и неизвестно, кто в каком доме живет. Однако известно, что:

- 1) Столяр живет правее Хирурга
- 2) Окулист живет левее Хирурга
- 3) Токарь живет с краю
- 4) Токарь живет рядом с Окулистом
- 5) Егор не Токарь и не живет рядом с Токарем
- 6) Михаил живет рядом с Хирургом
- 7) Алексей живет правее Окулиста
- 8) Алексей живет через дом от Михаила

Выясните, кто какой профессии и кто где живет, и дайте ответ в виде заглавных букв имени людей, в порядке слева направо. Например, если бы в домах жили (слева направо) Константин, Николай, Роман и Олег, ответ был бы: КНРО.

19. В школьном первенстве по настольному теннису в четверку лучших вошли девушки: Наташа, Маша, Люда и Рита. Самые горячие болельщики высказали свои предположения о распределении мест в дальнейших состязаниях. Один считает, что первой будет Наташа, а Маша будет второй. Другой болельщик на второе место прочит Люду, а Рита, по его мнению, займет четвертое место. Третий любитель тенниса с ними не согласился. Он считает, что Рита займет третье место, а Наташа будет второй. Когда соревнования закончились, оказалось, что каждый из болельщиков был прав только в одном из своих прогнозов. Какое место на чемпионате заняли Наташа, Маша, Люда, Рита? (В ответе перечислите подряд без пробелов числа, соответствующие местам девочек в указанном порядке имен.)

20. Когда сломался компьютер, его хозяин сказал: «Память не могла выйти из строя». Его сын предположил, что сгорел процессор, а винчестер исправен. Пришедший специалист по обслуживанию сказал, что, скорее всего, с процессором все в порядке, а память неисправна. В результате оказалось, что двое из них сказали все верно, а третий — все неверно. Что же сломалось?

21. Алеша, Витя и Игорь после уроков нашли на полу в кабинете физики маленькую гирьку. Каждый из них, рассматривая находку, высказал два предположения. Алеша сказал: «Это гирька из латуни, и весит она, скорей всего, 5 г», Витя предположил, что гирька сделана из меди и весит 3 г. Игорь же считал, что гирька не из латуни и вес ее — 4 г. Учитель физики обрадовался, что пропажа нашлась, и сказал ребятам, что каждый из них прав только наполовину. Из какого металла — латуни (Л) или

меди (М) — изготовлена гирька и каков ее вес? В ответе запишите первую букву названия металла, а затем цифру, соответствующую весу гирьки, например Л4.

22. На перекрестке произошло дорожно-транспортное происшествие, в котором участвовали автобус (А), грузовик (Г), легковой автомобиль (Л) и маршрутное такси (М). Свидетели происшествия дали показания инспектору ГИБДД. Первый свидетель считал, что первым на перекресток выехал автобус, а маршрутное такси было вторым. Другой свидетель полагал, что последним на перекресток выехал легковой автомобиль, а вторым был грузовик. Третий свидетель уверял, что автобус выехал на перекресток вторым, а следом за ним — легковой автомобиль. В результате оказалось, что каждый из свидетелей был прав только в одном из своих утверждений. В каком порядке выехали машины на перекресток? В ответе перечислите подряд без пробелов первые буквы названий транспортных средств в порядке их выезда на перекресток, например АМЛГ.

23. Три друга Олег, Борис и Арсений, окончив институт, разъехались по разным городам. И вот спустя несколько лет они, встретившись на вечере встречи выпускников, решили разыграть своего товарища. На его вопрос, где они теперь живут, друзья ответили:

Олег: Я живу в Екатеринбурге, а Борис — в Мурманске.

Борис: Я живу в Волгограде, а Олег — в Мурманске.

Арсений: Я живу в Мурманске, а Олег — в Волгограде.

Каждый из них один раз сказал правду и один раз солгал. Где живут Арсений, Борис и Олег? В ответе перечислите подряд без пробелов первые буквы названий городов, соответствующие именам друзей в указанном порядке, например ВМЕ.

Раздел 3

ОСНОВЫ АЛГОРИТМИЗАЦИИ

3.1. ОСНОВНЫЕ ПОНЯТИЯ

Алгоритм — заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий над данными для получения решения задачи за конечное число шагов.

Исполнитель алгоритма — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

Исполнителя характеризуют:

- среда;
- элементарные действия;
- система команд;
- отказы.

Среда (или обстановка) — это «место обитания» исполнителя. Например, для исполнителя Робота из задания A18 среда — это клеточное поле. Стены тоже часть среды, их расположение и положение самого Робота задают конкретное **состояние среды**.

Система команд. Каждый исполнитель может выполнять команды только из некоторого строго заданного списка — системы команд исполнителя. Для каждой команды должны быть заданы **условия применимости** (в каких состояниях среды может быть выполнена команда) и описаны **результаты выполнения команды**. Например, команда Робота «вверх» может быть выполнена, если выше Робота нет стены, ее результат — смещение Робота на одну клетку вверх. После вызова команды исполнитель совершает соответствующее **элементарное действие**.

Отказы исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды. Так, если выше Робота стена, то при попытке выполнить команду «вверх» Робот разрушится, и выполнение алгоритма прервется.

В информатике универсальным исполнителем алгоритмов является **компьютер**.

На практике наиболее распространены следующие формы представления алгоритмов:

- *словесная* (запись на естественном языке);
- *графическая* (изображения из блоков (геометрических фигур));
- *псевдокоды* (описание алгоритмов на условном языке с русским или английским синтаксисом);
- *программная* (тексты на языках программирования).

Остановимся подробнее на первых двух формах.

3.2. СЛОВЕСНЫЙ СПОСОБ ПРЕДСТАВЛЕНИЯ АЛГОРИТМА

Этот способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.

Пример 1. Записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел (алгоритм Эвклида).

Решение. Алгоритм может быть следующим:

- задать два числа;
- если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
- определить большее из чисел;
- заменить большее из чисел разностью большего и меньшего из чисел;
- повторить алгоритм с шага 2.

Описанный алгоритм применим к любым натуральным числам и должен приводить к решению поставленной задачи.

Пример 2. Саша забыл пароль для запуска компьютера, но помнил алгоритм его получения из строки подсказки «QQ3QWER»: если все последовательности символов «QQ» заменить на «QWER», а затем из получившейся строки удалить сочетания символов «3Q», то полученная последовательность и будет паролем. Определите пароль:

- | | |
|-------------|------------|
| 1) QWERQWER | 3) QWERWER |
| 2) QWER 3QQ | 4) QWERQ |

Решение. В результате выполнения действия «последовательности символов «QQ» заменить на «QWER» получим последовательность QWER3QWER. А после выполнения действия «удалить сочетания символов «3Q» останется последовательность QWERWER.

Верный ответ: 3).

Пример 3. В слове «программирование» уничтожается каждый третий символ, затем слово переворачивается и опять уничтожается каждый второй символ. Эти действия повторяются до тех пор, пока не останется два символа. В ответе укажите эти два символа.

Решение. Вычеркнем каждую третью букву из первоначального слова «ПРОГРАММИРОВАНИЕ». Перевернем оставшееся слово и вычеркнем в нем каждую третью букву «ЕНАОРММРФРП». Еще раз перевернем и вычеркнем: «ПРРМРӨНЕ». Повторим те же самые действия: «ЕНРМРН». Далее получаем «РМНЕ». И в итоге «ЕМР».

Верный ответ: ЕМ.

Пример 4. Известно, что в строке, состоящей из 10 символов, есть один символ X и 9 символов Y. В этой строке стали вычеркивать символы по следующим правилам (счет начали с начала строки):

1. Зачеркивается каждый третий символ.
2. При достижении конца строки счет продолжается с начала строки.
3. Уже зачеркнутые символы не считаются.

Последним незачеркнутым символом оказался символ Х. Каким по счету он стоял в строке?

Решение. Не будем гадать о месте символа Х. Занесем в таблицу только символы У, затем начнем вычеркивать каждый третий символ. При переходе на следующую строку (то есть при возврате к началу строки) не будем переписывать зачеркнутые символы, чтобы не загромождать запись.

1	2	3	4	5	6	7	8	9	10
У	У	У	У	У	У	У	У	У	У
У	У		У	У		У	У		У
У			У	У			У		У
			У	У					У
			У						У
			Х						

В итоге остался 4-й символ, это и есть место, на котором находится символ Х.

Верный ответ: 4.

Пример 5. У исполнителя Удвоитель две команды, которым присвоены номера:

1. вычти 2,
2. умножь на 3.

Первая из них уменьшает число на экране на 2, вторая — утраивает его. Запишите порядок команд в программе *получения из 3 числа 19, содержащей не более 4 команд*, указывая лишь номера команд. (Например, программа **221** — это программа:

умножь на 3, умножь на 3, вычти 2, которая преобразует число 1 в 7.).

Решение. Порядок команд в данном случае удобнее определять с конца. 19 — не кратно 3, значит, последним действием было вычитание цифры 2. Получаем число 21, кратное 3. Вероятнее всего, предпоследним действием было умножение на 3. Рассуждая аналогичным образом, получаем последовательность действий:

19 ← 21 ← 7 ← 9 ← 3.

1 2 1 2

Меняя порядок действий, получаем 2121.

Верный ответ: 2121.

Пример 6. Алгоритм обработки числовой последовательности задан следующим образом:

1. Все элементы последовательности обрабатываются поочередно слева направо, начиная с первого и заканчивая последним элементом.

2. Если текущий элемент последовательности — нечетное число, из него вычитается 3.

3. Если текущий элемент последовательности — четное число, из него вычитается 1.

4. Если текущий элемент последовательности — ноль, к нему прибавляется 2.

Этот алгоритм три раза последовательно применили к следующей числовой последовательности: 12345432123454321.

Сколько в результирующей последовательности элементов отрицательных чисел? В ответе укажите целое число.

Решение. Выполним вышеперечисленные требования и занесем результаты в таблицу.

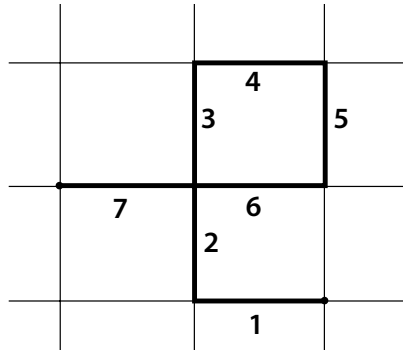
	1	2	3	4	5	4	3	2	1	2	3	4	5	4	3	2	1
1)	-2	1	0	3	2	3	0	1	-2	1	0	3	2	3	0	1	-2
2)	-3	-2	2	0	1	0	2	-2	-3	-2	2	0	1	0	2	-2	-3
3)	-6	-3	1	2	-2	2	1	-3	-6	-3	1	2	-2	2	1	-3	-6

В последней строке 9 отрицательных чисел.

Верный ответ: 9.

Пример 7. Исполнитель Робот ходит по клеткам бесконечной вертикальной клетчатой доски, переходя по одной из команд **вверх**, **вниз**, **вправо**, **влево** в соседнюю клетку в указанном направлении. Робот выполнил следующую программу:

1. влево
2. вверх
3. вверх
4. вправо
5. вниз
6. влево
7. влево



Укажите наименьшее возможное число команд в программе, приводящей Робота из той же начальной клетки в ту же конечную.

Решение. Выполним программу, составленную для Робота.

Попасть в исходную клетку Робот может несколькими путями: например, **вправо, вправо, вниз** или **вниз, вправо, вправо**. В любом случае вариантов число команд будет не менее трех.

Верный ответ: 3.

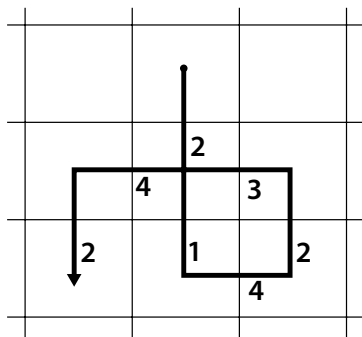
Замечание. Иногда условие задачи усложняется наличием стен между клетками, в этом случае возвращаться нужно по тому же пути, по которому Робот проходил свой путь. Рассмотрим следующий пример.

Пример 8. 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) и 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

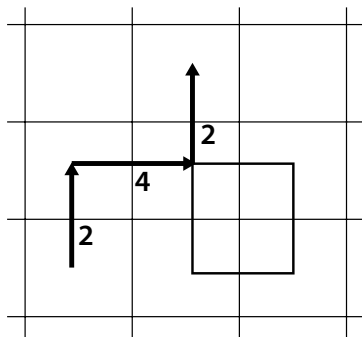
2324142

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. Изобразим путь, который прошел Робот. Вернуться в начальную клетку нужно по уже пройденному маршруту, так как расположение стен между клетками неизвестно.



Верный ответ: 242.

Пример 9. Была исходная последовательность символов: ААВВАВВ. В конец этой последовательности дописали ее копию, но развернутую зеркально (символы взяли в обратном порядке). Получилась строка: АААВВВАВВВВАВВВААА.

Эту операцию повторили еще три раза, каждый раз дописывая в зеркальном отображении всю последовательность, полученную на предыдущем шаге. В результате получилась последовательность из 128 символов. В получив-

шейся последовательности заменили все тройки идущих подряд символов BAV на ABA . Эту операцию повторяли до тех пор, пока тройки идущих подряд символов BAV не перестали встречаться в последовательности. Сколько букв B осталось в результирующей последовательности?

Решение. Заметим, что строка $AAABVBAVVVBAVBAAAA$ является палиндромом, то есть читается одинаково слева направо и справа налево. А значит, при зеркальном развороте получаем ту же самую строку, в которой 16 символов. Итоговая последовательность состоит из $128 : 16 = 8$ идущих подряд строк $AAABVBAVVVBAVBAAAA$, поэтому проделаем все операции с этой строкой, а затем результат умножим на 8. Это возможно в силу того, что начало и конец строки не содержат символов B .

Выполним замены:

строка $AAABVBAVVVBAVBAAAA \Rightarrow$ в строку
 $AAABAVABVBAVBAAAA$

строка $AAABAVABVBAVBAAAA \Rightarrow$ в строку
 $AAAABAABAABAABA$

строка $AAAABAABAABAABA \Rightarrow$ в строку
 $AAAABAABAABAABA.$

Больше символы BAV в строке не встречаются. В строке 3 символа B . Умножим это число на 8 (количество строк в последовательности).

Верный ответ: 24.

Пример 10. Цепочки символов (строки) создаются по следующему правилу. Первая строка состоит из одного символа — цифры «1». Каждая из последующих цепочек создается следующим действием: в очередную строку дважды записывается предыдущая цепочка цифр (одна за другой, подряд), а в конец приписывается еще одно число — номер строки по порядку (на i -м шаге дописывается число « i »).

Вот первые 4 строки, созданные по этому правилу:

(1) 1

(2) 112

(3) 1121123

(4) 112112311211234

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

Решение. Обозначим через K_i количество четных цифр в i -й строке. В первой строке нет четных цифр, значит, $K_1 = 0$. Во второй строке 1 четное число «2», то есть $K_2 = 1$. В третьей строке удваивается предыдущая строка, а значит, удваивается и количество четных цифр $K_3 = 2 \cdot K_2 = 2$. В четвертой строке тоже происходит увеличение в 2 раза и еще добавляется 1 цифра «4»: $K_4 = 2 \cdot K_3 + 1 = 5$. Рассуждая подобным образом далее, получаем $K_5 = 2 \cdot K_4 = 10$, $K_6 = 2 \cdot K_5 + 1 = 21$, $K_7 = 2 \cdot K_6 = 42$, $K_8 = 2 \cdot K_7 + 1 = 85$.

Верный ответ: 85.

Пример 11. Строки (цепочки 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 слева дважды подряд приписывается предыдущая строка.

Вот первые 4 строки, созданные по этому правилу:

- (1) А
- (2) ААВ
- (3) ААВААВС
- (4) ААВААВСААВААВСD

Латинский алфавит (для справки):

АВСDEFGHIJKLMNOPQRSTUVWXYZ

Запишите шесть символов подряд, стоящие в седьмой строке со 117-го по 122-е место (считая слева направо).

Решение. Посчитаем количество символов в каждой строке:

- (1) 1
- (2) $2 \cdot 1 + 1 = 3$
- (3) $2 \cdot 3 + 1 = 7$
- (4) $2 \cdot 7 + 1 = 15$
- (5) $2 \cdot 15 + 1 = 31$
- (6) $2 \cdot 31 + 1 = 63$
- (7) $2 \cdot 63 + 1 = 127$

Таким образом, в 7-й строке 127 символов, причем последние 7 идут в алфавитном порядке (выделенная часть каждой строки в постановке задачи):

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

Символы, стоящие левее 121-го, можно взять, например, из 4-й строки

...A A B A A B ...
...117 118 119 120 121 122 ...

Верный ответ: AABAAB.

Пример 12. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

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

При выполнении этих команд РОБОТ перемещается на одну клетку соответственно: вверх, вниз, влево, вправо.

Четыре команды служат для проверки истинности условия отсутствия соответствующей стены у той клетки, где находится РОБОТ:

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

Цикл ПОКА *< условие >* команда выполняется, пока условие истинно, иначе происходит переход на следующую строку.

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

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

НАЧАЛО

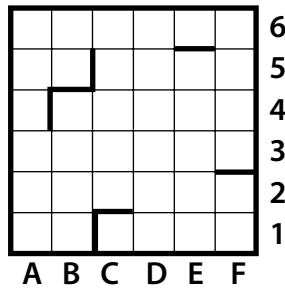
ПОКА **< сверху свободно >** **вправо**

ПОКА **< справа свободно >** **вниз**

ПОКА **< снизу свободно >** **влево**

ПОКА **< слева свободно >** **вверх**

КОНЕЦ

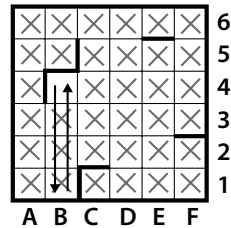
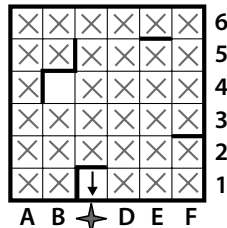
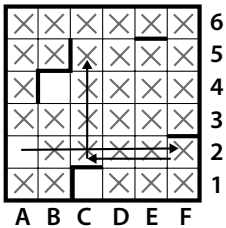
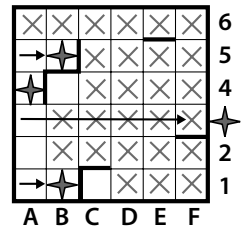
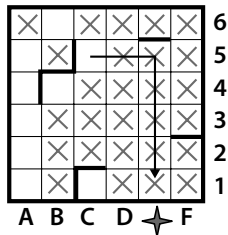
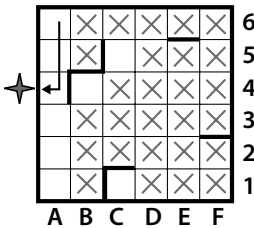


- 1) 1 2) 2 3) 3 4) 4

Решение.

Заметим, что выполнение алгоритма заканчивается тогда, когда условие **< слева свободно >** становится ложным. Так как нас интересуют клетки, в которых начинается и заканчивается движение Робота, то понятно, что слева у таких клеток должна быть стена. Вычеркнем все клетки, не удовлетворяющие этому условию.

Для оставшихся клеток лабиринта выполним **предложенный алгоритм**. Звездочкой показано разрушение Робота.



На последнем рисунке крестиками помечены клетки, не подходящие под требование задачи, а кружком — клетка, удовлетворяющая условию задачи.

×	×	×	×	×	×	6
×	×	×	×	×	×	5
×	○	×	×	×	×	4
×	×	×	×	×	×	3
×	×	×	×	×	×	2
×	×	×	×	×	×	1
A	B	C	D	E	F	

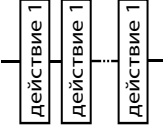
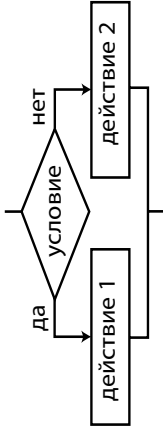
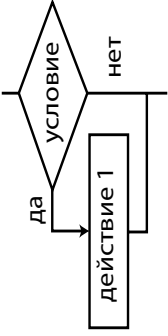
Верный ответ: 1).

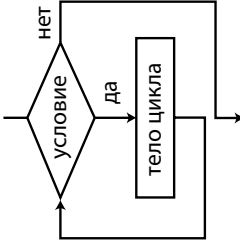
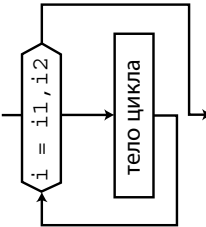
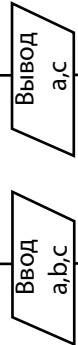

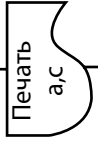
Словесный способ не имеет широкого распространения, так как такие описания:

- строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

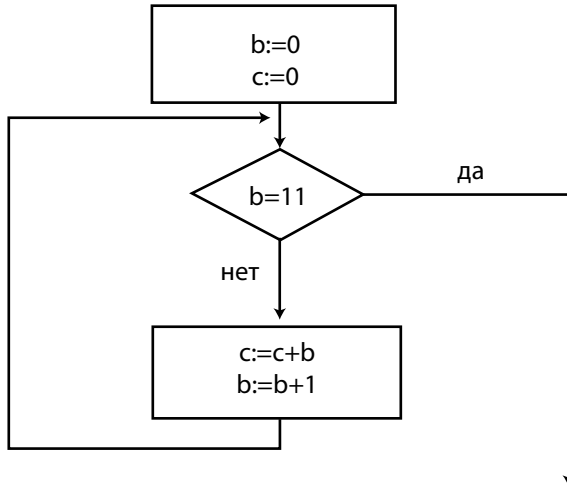
3.3. ГРАФИЧЕСКИЙ СПОСОБ ПРЕДСТАВЛЕНИЯ АЛГОРИТМА

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным. Такое графическое представление называется схемой алгоритма или **блок-схемой**. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде **блочного символа**. Блочные символы соединяются **линиями переходов**, определяющими очередность выполнения действий. В таблице приведены наиболее часто употребляемые символы.

Название символа	Обозначение и пример заполнения	Пояснение
Следование		Вычислительное действие или последовательность действий
Ветвление	<p>Полная форма ветвления</p>  <p>Неполная форма ветвления</p> 	Обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран

<p>Цикл с предусловием</p>		<p>Предписывает повторять тело цикла до тех пор, пока выполняется условие, записанное в начале циклической конструкции</p>
<p>Цикл с параметром</p>		<p>Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне</p>
<p>Ввод и вывод данных</p>		<p>Ввод данных с клавиатуры в общем виде</p>
<p>Пуск-останов</p>		<p>Начало, конец алгоритма, вход и выход в подпрограмму</p>
<p>Вывод результата на печать</p>		<p>Вывод результатов на печать</p>

Пример 13. Определите значение переменной c после выполнения фрагмента алгоритма:



Примечание: знаком $:=$ обозначена операция присваивания.

Решение. В данной блок-схеме представлены две алгоритмические конструкции: следование и цикл с предусловием. Для определения значения переменной c выполним алгоритм пошагово и оформим этот процесс в виде таблицы.

Номер шага	Проверка условия $b = 11$	Значение переменной c	Значение переменной b
0	-----	0	0
1	Неверно	$0 + 0 = 0$	$0 + 1 = 1$
2	Неверно	$0 + 1 = 1$	$1 + 1 = 2$
3	Неверно	$1 + 2 = 3$	$2 + 1 = 3$
4	Неверно	$3 + 3 = 6$	$3 + 1 = 4$
5	Неверно	$6 + 4 = 10$	$4 + 1 = 5$
6	Неверно	$10 + 5 = 15$	$5 + 1 = 6$
7	Неверно	$15 + 6 = 21$	$6 + 1 = 7$
8	Неверно	$21 + 7 = 28$	$7 + 1 = 8$
9	Неверно	$28 + 8 = 36$	$8 + 1 = 9$

Номер шага	Проверка условия $b = 11$	Значение переменной c	Значение переменной b
10	Неверно	$36 + 9 = 45$	$9 + 1 = 10$
11	Неверно	$45 + 10 = 55$	$10 + 1 = 11$
12	Верно	-----	-----

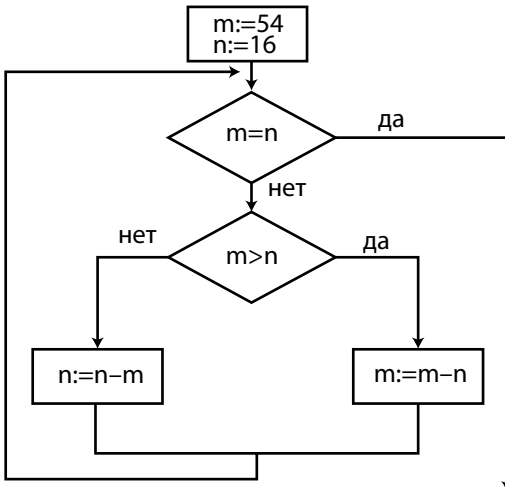
Верный ответ: 55.

Замечание. Для человека, неплохо знакомого с математикой, несложно увидеть, что в блок-схеме заложен алгоритм вычисления суммы первых 10 членов арифметической прогрессии:

$$\begin{aligned}
 S &= \frac{a_1 + a_n}{2} \cdot n = \frac{a_1 + a_1 + (n-1) \cdot d}{2} \cdot n = \\
 &= \frac{1 + 1 + (10-1) \cdot 1}{2} \cdot 10 = \frac{11}{2} \cdot 10 = 55.
 \end{aligned}$$

Это может служить дополнительной проверкой при решении задачи В2.

Пример 14. Определите значение переменной m после выполнения фрагмента алгоритма.

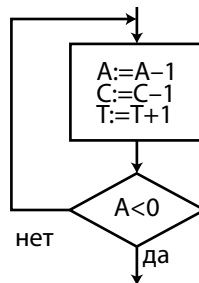


Решение. В приведенной блок-схеме представлены три алгоритмические **конструкции**: следование, цикл с пред-условием и ветвление в теле цикла. Для определения значения переменной m выполним алгоритм пошагово. Заметим, что блок-схема описывает алгоритм Евклида нахождения наибольшего общего делителя двух чисел (пример 1). Запомните его, он наверняка пригодится вам в дальнейшем.

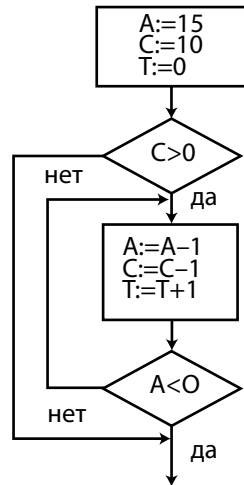
Номер шага	Проверка условия $m = n$	Проверка условия $m > n$	Значение переменной m	Значение переменной n
0	-----	-----	54	16
1	Неверно	Верно	$54 - 16 = 38$	16
2	Неверно	Верно	$38 - 16 = 22$	16
3	Неверно	Верно	$22 - 16 = 6$	16
4	Неверно	Неверно	6	$16 - 6 = 10$
5	Неверно	Неверно	6	$10 - 6 = 4$
6	Неверно	Верно	$6 - 4 = 2$	4
7	Неверно	Неверно	2	$4 - 2 = 2$
8	Верно	-----	-----	-----

Верный ответ: 2.

Пример 15. Дана блок-схема алгоритма. Какое значение приобретет переменная T после завершения выполнения алгоритма?



Решение. Первоначально переменная $C = 10$, то есть больше нуля. Далее попадаем внутрь конструкции. Для того чтобы переменная A стала отрицательной, необходимо отнять от A 16 раз единицу. Соответственно на 16 станет меньше переменная C ($C = 10 - 16 = -6$) и на 16 увеличится переменная T ($T = 0 + 16 = 16$). Таким образом, $T = 16$.



Верный ответ: 16.

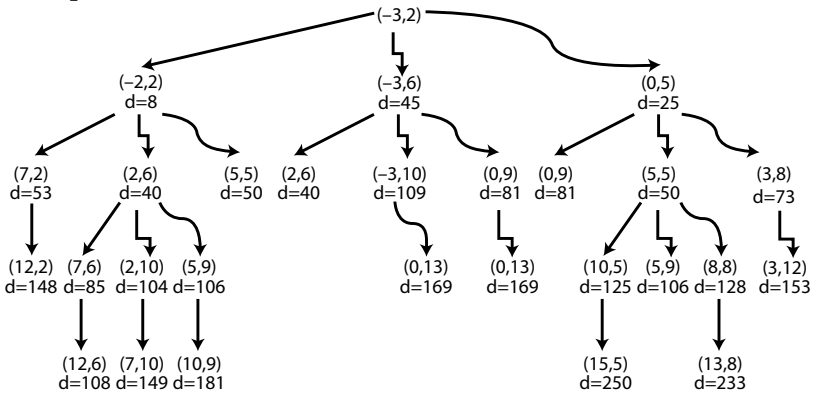
3.4. СПОСОБ ЗАПИСИ АЛГОРИТМА В ВИДЕ ГРАФА

При решении задачи СЗ удобно использовать дерево вариантов.

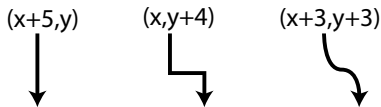
Пример 16. Два игрока играют в следующую игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. В начале игры фишка находится в точке с координатами $(-3, 2)$. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: или в точку с координатами $(x + 5, y)$, или в точку с координатами $(x, y + 4)$, или в точку с координатами $(x + 3, y + 3)$. Выигрывает игрок, после хода которого расстояние по прямой от фишки до точки с координатами $(0, 0)$ больше 12 единиц. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Решение. Игра заканчивается, если квадрат расстояния $d = R^2 \geq 144$. Построим граф вариантов.

Приведенное дерево вариантов содержит не все варианты ходов: не прописаны дублирующие ходы. Иногда указан только один из трех ходов, после которого игрок выигрывает.

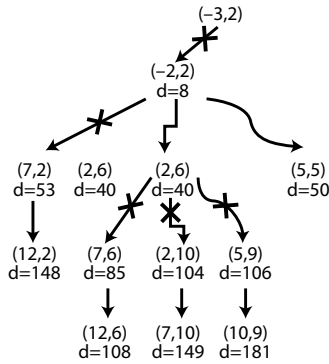
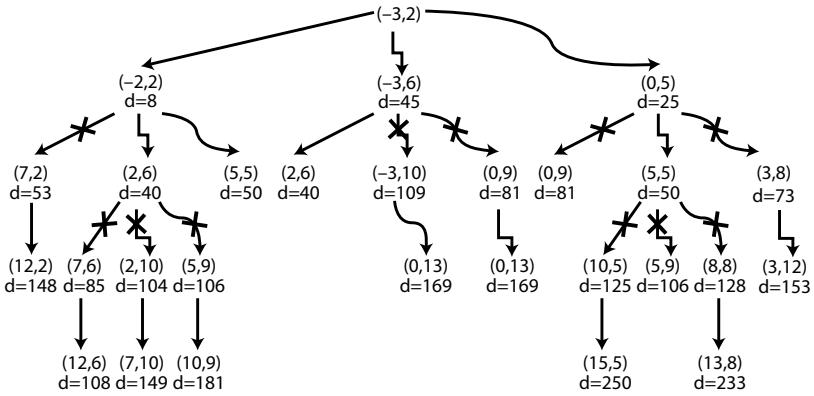


При построении графа ходы I игрока записаны черным цветом, а ходы II игрока — серым цветом. Это сделано для того, чтобы не запутаться. Советуем вам использовать ручки с чернилами разного цвета. Кроме того, различный тип линий соответствует разным вариантам хода:

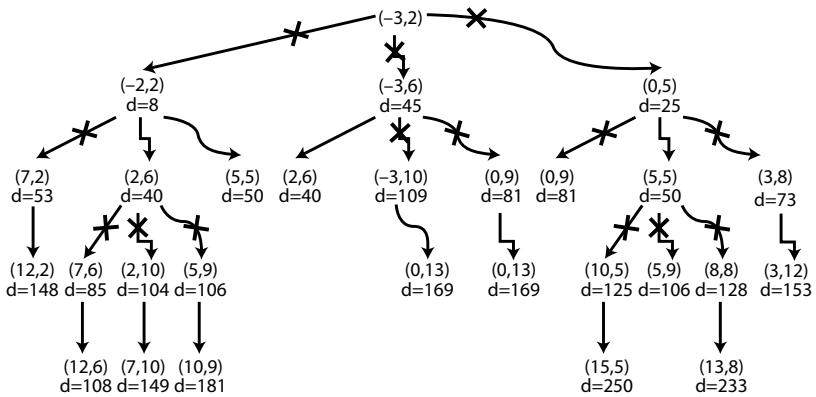


Далее вычеркнем ходы, не выгодные для игрока, — это те ходы, в результате которых следующим ходом выигрывает противник.

Важное замечание. Если на каком-либо ходе для какого-либо игрока не выгоден ни один ход, то соответственно не выгоден и предыдущий ход данного игрока. То есть из приведенного здесь фрагмента следует, что предыдущий ход I игрока не является выигрышным, так как II игрок в данных обстоятельствах может сделать ход (2,6), после которого у I игрока не остается шансов на победу. Рассуждая подобным образом, видим, что ходы (-3,6) и (0,5) I игрока тоже не ведут к выигрышу.



Отразим это на дереве вариантов:



На основе графа построена таблица стратегий. Обратите внимание, что для выигрывающего игрока указаны только выигрышные ходы, а для проигрывающего игрока указаны все варианты ходов.

Строго говоря, для обоснованного ответа на сформулированный в задаче вопрос достаточно:

- записать таблицу вариантов;
- указать выигрывающего игрока;
- указать те шаги, которые ведут выигрывающего игрока к победе.

Дерево вариантов строится только на черновике.

	I ход	II ход	III ход	IV ход
Стартовая позиция	I игрок (все варианты)	II игрок (выигрышный ход)	I игрок (все варианты ходов)	II игрок (выигрышный ход, один из вариантов)
(-3,2)	(2,2)	<u>(2,6)</u>	(7,6)	<u>(12,6)</u>
			(2,10)	<u>(7,10)</u>
			(5,9)	<u>(10,9)</u>
		<u>(5,5)</u>	(5,9)	<u>(10,9)</u>
			(10,5)	<u>(15,5)</u>
			(8,8)	<u>(13,8)</u>
	(-3,6)	<u>(2,6)</u>
		(0,5)	<u>(5,5)</u>	...

Ответ: выиграет II игрок, так как при любом ходе I игрока II игрок имеет выигрышный ход, приведенный и подчеркнутый в таблице. Первый ход II игрока должен быть таким, чтобы фишка попала в точку (2,6) или (5,5).

Задача СЗ может иметь и другую формулировку, но способ решения при этом не меняется. Однако внимательно читайте формулировку задачи, так как в ней могут быть сформулированы различные варианты выигрыша или проигрыша игроков. Приведем в качестве примера еще одну задачу СЗ и несколько другой способ оформления дерева вариантов.

1			8,3					4,6					4,7				
2	4,3		8,6		8,7		12,3	<u>8,6</u>	<u>4,12</u>	4,10	8,7	<u>4,14</u>	<u>4,11</u>				
1	16,6 20,3 16,7 16,6 8,12 12,6 8,10 16,7 8,11 12,7 8,11																
2			12,12 8,14		12,11 12,10 16,7 16,7					8,14 8,14					12,11 8,15		
	победа 1-го игрока	победа 1-го игрока	победа 2-го игрока	победа 1-го игрока	победа 2-го игрока победа 2-го игрока победа 2-го игрока победа 2-го игрока	победа 1-го игрока	победа 1-го игрока	победа 2-го игрока	победа 1-го игрока	победа 1-го игрока	победа 1-го игрока	победа 1-го игрока	победа 1-го игрока	победа 2-го игрока			

Построим таблицу стратегий, в которой для выигрывающего игрока указаны только выигрышные ходы, а для проигрывающего игрока указаны все варианты ходов.

	I ход	II ход	III ход	IV ход
Стартовая позиция	I игрок (все варианты ходов)	II игрок (выигрышный ход)	I игрок (все варианты ходов)	II игрок (выигрышный ход)
4,3	8,3	<u>12,3</u>	12,6	<u>16,6</u>
			12,7	<u>12,11</u>
			16,3	<u>16,6</u>
	4,6	<u>4,10</u>	8,10	<u>8,14</u>
			4,14	<u>8,14</u>
	4,7	<u>4,11</u>	8,11	<u>12,11</u>
4,15			<u>8,15</u>	

Ответ: выиграет II игрок, так как при любом ходе I игрока II игрок имеет выигрышный ход, приведенный и подчеркнутый в таблице.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Основы алгоритмизации

1. Витя пригласил своего друга Сергея в гости, но не сказал ему код от цифрового замка своего подъезда, а послал следующее SMS-сообщение: «В последовательности чисел 3, 1, 8, 2, 6 все числа больше 5 разделить на 2, а затем удалить из полученной последовательности все четные числа». Выполнив указанные в сообщении действия, Сергей получил следующий код для цифрового замка:

- | | |
|------------|------------|
| 1) 3, 1 | 3) 3, 1, 3 |
| 2) 1, 1, 3 | 4) 3, 3, 1 |

2. Маша забыла пароль для входа в Windows XP, но помнила алгоритм его получения из символов «КВМAM9КВК» в строке подсказки. Если все последовательности символов «МAM» заменить на «RP», а «КВК» — на «1212», а из получившейся строки удалить 3 последние символа, то полученная последовательность и будет паролем.

- | | |
|-------------|------------|
| 1) KBRP91 | 3) KBRP |
| 2) 1212RP91 | 4) KB91212 |

3. Юлия забыла пароль для входа в Windows XP, но помнила алгоритм его получения из символов «QWER3QWER1» в строке подсказки. Если все последовательности символов «QWER» заменить на «QQ», а из получившейся строки удалить последовательность «3Q», то полученная последовательность и будет паролем:

- | | |
|-------------|-----------|
| 1) 3QQQ1 | 3) QQWER1 |
| 2) QWERWER1 | 4) QQQ1 |

4. Алексей пригласил своего друга Володю в гости, но не сказал ему код от цифрового замка своего подъезда, а послал следующее SMS-сообщение: «В последовательности цифр 7, 1, 8, 3, 4 все цифры меньше 5 увеличить на 1, а затем удалить из полученной последовательности все нечетные цифры». Выполнив действия, указанные в сообщении, Володя получил следующий код для цифрового замка:

- 1) 84 2) 284 3) 482 4) 824

5. 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

6. Исполнитель Чертежник имеет перо, которое можно поднимать, опускать и перемещать. При перемещении опущенного пера за ним остается след в виде прямой линии. У исполнителя существуют следующие команды:

Сместиться на вектор (a, b) — исполнитель перемещается в точку, в которую можно попасть из данной, пройдя a единиц по горизонтали и b — по вертикали.

Запись: **Повторить 5 [Команда1 Команда2]** означает, что последовательность команд в квадратных скобках повторяется 5 раз.

Чертежник находится в начале координат. Чертежнику дан для исполнения следующий алгоритм:

Сместиться на вектор $(5, 2)$

Сместиться на вектор $(-3, 3)$

Повторить 3 [Сместиться на вектор $(1, 0)$]

Сместиться на вектор $(3, 1)$

На каком расстоянии от начала координат будет находиться исполнитель Чертежник в результате выполнения данного алгоритма?

7. 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?

8. У исполнителя Калькулятор две команды, которым присвоены номера:

- 1) прибавь 2
- 2) умножь на 3

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

- умножь на 3
- прибавь 2
- умножь на 3
- прибавь 2
- прибавь 2,

которая преобразует число 1 в 19.)

9. У исполнителя Утроитель 2 команды, которым присвоены номера:

- 1) вычти 2
- 2) умножь на три

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

умножь на три
 вычти 2
 умножь на три
 вычти 2
 вычти 2,
 которая преобразует число 2 в 8.)

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

10. Исполнитель КАЛЬКУЛЯТОР имеет только две команды, которым присвоены номера:

- 1) умножь на 2
- 2) вычти 1

Выполняя команду номер 1, КАЛЬКУЛЯТОР умножает число на экране на 2, а выполняя команду номер 2, вычитает из числа на экране 1. Напишите программу, содержащую не более 4 команд, которая из числа 7 получает число 52. Укажите лишь номера команд.

Например, программа 12121 — это программа:

умножь на 2
 вычти 1
 умножь на 2
 вычти 1
 умножь на 2,
 которая преобразует число 5 в число 34.

11. Цепочки символов (строки) создаются по следующему правилу.

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

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

Вот первые 4 строки, созданные по этому правилу:

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

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

12. Строки (цепочки символов латинских букв) создаются по следующему правилу. Нулевая цепочка состоит из одного символа — цифры «0». Каждая из последующих цепочек создается такими действиями: в начале дважды подряд записывается предыдущая строка, а затем — инвертированная предыдущая строка (в которой цифра «0» заменяется на «1», и наоборот). Вот первые 4 строки, созданные по этому правилу:

(0) 0

(1) 001

(2) 001001110

(3) 001001110001001110110110001

Сколько раз встречается цифра «1» в строке с номером 6?

13. Записано 6 строк, каждая имеет свой номер — от «0» до «5». В «0»-й 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)?

14. Цепочки символов (строки) создаются по следующему правилу: первая строка состоит из одного символа — цифры «1». Каждая из последующих цепочек создается такими действиями: в начало записывается число — номер строки по порядку (для i -й строки ставится число « i »), далее дважды подряд записывается предыдущая строка. Вот первые 4 строки, созданные по этому правилу:

- (1) 1
- (2) 211
- (3) 3211211
- (4) 432112113211211

Сколько раз встречается цифра «1» в первых семи строках (суммарно)?

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

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

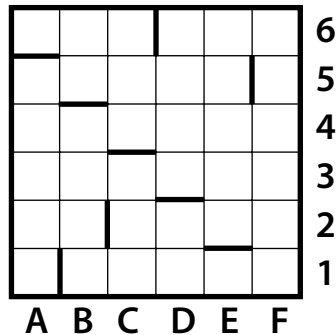
При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх \uparrow , вниз \downarrow , влево \leftarrow , вправо \rightarrow .

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 \langle условие \rangle команда

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



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

НАЧАЛО

ПОКА < снизу свободно > вниз

ПОКА < слева свободно > влево

ПОКА < сверху свободно > вверх

ПОКА < справа свободно > вправо

КОНЕЦ

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

16. 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 →.

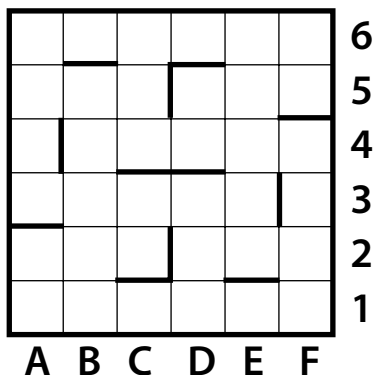
Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

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 > вверх

ПОКА < сверху свободно > влево

ПОКА < слева свободно > вниз

ПОКА < снизу свободно > вправо

КОНЕЦ

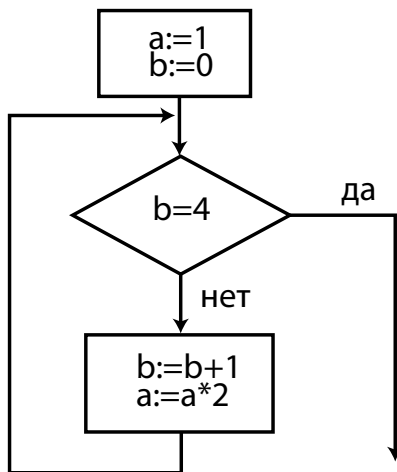
1) 1

2) 2

3) 3

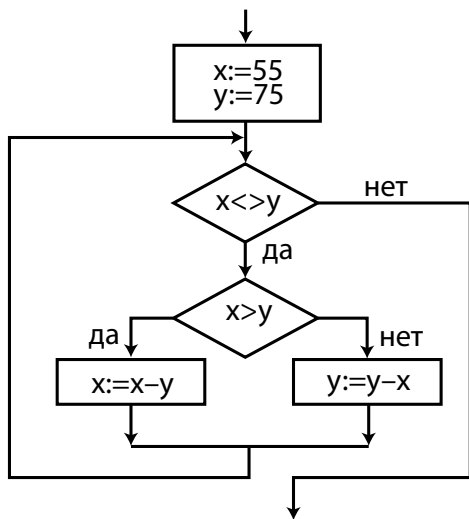
4) 4

17. Определите значение переменной a после выполнения фрагмента алгоритма:

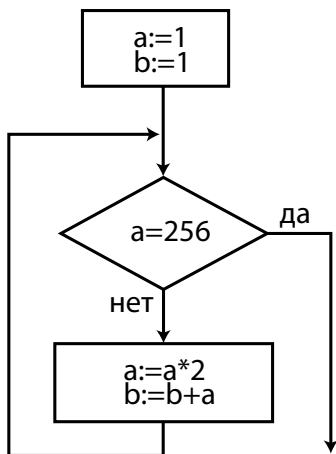


Примечание: знаком $*$ обозначено умножение, знаком $:=$ обозначена операция присваивания.

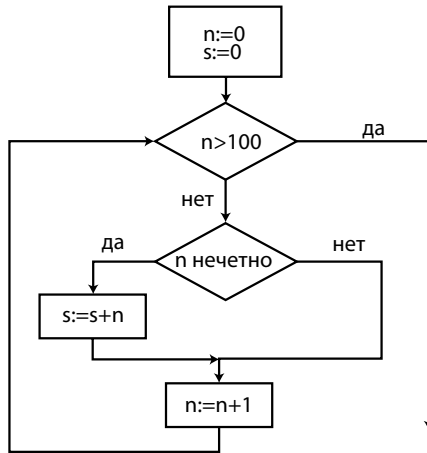
18. Определите значение целочисленной переменной x после выполнения следующего фрагмента программы:



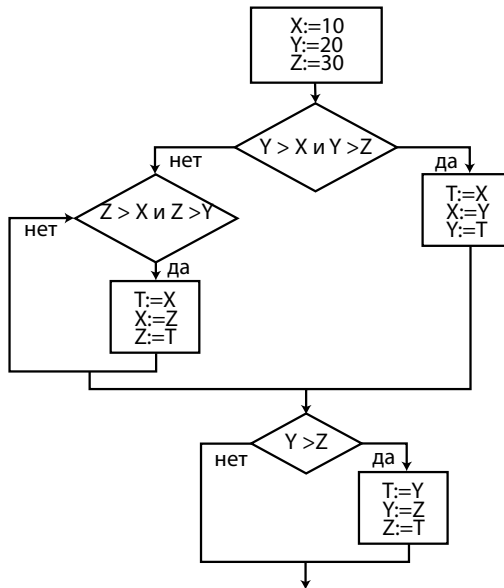
19. Запишите значение переменной b после выполнения фрагмента алгоритма:



20. Запишите значение переменной s после выполнения фрагмента алгоритма:



21. Дана блок-схема алгоритма. Чему будут равны значения переменных X , Y и Z после его выполнения? В ответе укажите через запятую три числа: сначала значение X , затем значение Y и затем значение Z .



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

23. 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.

24. 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 увеличивает в 2 раза 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? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Раздел 4

ПРОГРАММИРОВАНИЕ

4.1. ОСОБЕННОСТИ ЯЗЫКА ПАСКАЛЬ

Язык программирования Pascal был разработан в 1968–1971 гг. Никлаусом Виртом в Цюрихском Институте информатики (Швейцария). Первоначально цель разработки языка диктовалась необходимостью разработки инструмента для обучения программированию. Однако очень скоро обнаружилась чрезвычайная эффективность применения языка Pascal для решения самых разнообразных задач. К настоящему времени Pascal принадлежит к группе наиболее распространенных и популярных в мире языков программирования.

Основными особенностями языка Pascal являются:

1. Программа на языке Pascal представляет собой специально организованную последовательность шагов по преобразованию данных, приводящую к решению задачи.

2. Язык Pascal содержит удобные средства для представления данных. Развитая система типов позволяет адекватно описывать данные, подлежащие обработке. Pascal является типизированным языком, что означает фиксацию типов переменных при их описании, а также строгий контроль преобразования типов и контроль доступа к данным в соответствии с их типом.

3. Набор операторов языка Pascal отражает принципы структурного программирования и позволяет записывать достаточно сложные алгоритмы в компактной и элегантной форме. Pascal является процедурным языком с традиционной блочной структурой и статически определенными областями действия имен.

4. Синтаксис языка несложен. Программы записываются в свободном формате, что позволяет сделать их наглядными и удобными для изучения.

4.2. ОБЩАЯ СТРУКТУРА ПРОГРАММ НА ЯЗЫКЕ ПАСКАЛЬ

Программы на Паскале имеют следующий общий вид:

```
Program Имя программы;  
Раздел описаний;  
Begin  
Раздел операторов;  
End.
```

Слова PROGRAM, BEGIN и END выделяют 2 части программы: раздел описаний и раздел операторов. Такая структура обязательна. Любой объект, используемый в программе, должен быть учтен в разделе описаний. Иными словами, в разделе описаний должны быть перечислены имена всех используемых в программе меток, констант, нестандартных типов данных, переменных. Кроме того, в разделе описаний приводятся тексты функций и процедур, определяемых пользователем. Раздел операторов — это основная часть программы, здесь содержатся команды, составляющие программу.

Программа, написанная по правилам стандартного языка Паскаль, должна иметь в своем полном варианте (когда в ней задействованы все виды программных объектов) следующую структуру:

```
Program <имя программы>;  
Uses <список используемых модулей>;  
Label <Список меток из основного блока программы>;  
Const <Определение констант программы>;  
Type <Описание типов>;  
Var <Описание переменных>;  
Procedure <Текст процедуры>;  
Function <Текст функции>;  
Begin  
<Основной блок программы>;  
(раздел операторов)  
End.
```

Некоторые из перечисленных блоков раздела описаний могут отсутствовать за ненадобностью, но оставшиеся

должны, по возможности, следовать в указанной последовательности.

Пример 1. Известен радиус окружности R . Требуется составить программу, которая будет вычислять длину окружности C и площадь круга S .

Решение. Для решения задачи требуется:

- ввести с клавиатуры величину радиуса окружности;
- вычислить длину окружности и площадь круга по формулам $C = 2\pi R$, $S = \pi R^2$;
- вывести результат на экран.

Заметим, что константа π в языке Паскаль является функцией без аргумента `pi`. Текст программы следующий.

```
program P1;
var R, C, S : real;
begin
writeln('введите радиус окружности');
readln(R);
C:=2*pi*R;
S:=pi*sqr(R);
writeln('C=',C:4:1,' S=',S:4:1);
readln;
end.
```

Описание оператора присваивания и процедур ввода-вывода смотрите в разделе 4.6, описание арифметических функций и стандартных математических функций — в разделе 4.5.

Стоящий в конце программы оператор **Readln без параметра** позволяет программным путем задержать окно пользователя USER SCREEN (окно ввода-вывода) до тех пор, пока не будет нажата клавиша ENTER.

4.3. АЛФАВИТ И ИДЕНТИФИКАТОРЫ

Алфавит языка Турбо-Паскаль состоит из полного набора символов клавиатуры, при этом *не различаются строчные и прописные буквы* (компилятору безразлична

высота букв). Пробел в Турбо-Паскале используется в качестве разделителя отдельных конструкций языка, поэтому слитное написание текста будет ошибочным.

Идентификатор — это имя объекта программы. Объектами являются сама программа, модули, процедуры, функции, типы данных, метки, константы, переменные. К идентификаторам (именам) всех перечисленных объектов предъявляются те же требования, что и к имени переменной (см. раздел 4.4).

Так, в примере 1 использованы идентификаторы P1, R, C, S.

Нельзя использовать один и тот же идентификатор для разных объектов. Например, нельзя использовать в программе переменную с именем, которое уже присвоено ранее какой-нибудь константе, процедуре либо другой переменной.

Не допускается использовать в качестве идентификаторов служебные слова, зарезервированные в языке Паскаль для других целей, например названия стандартных функций и процедур Паскаля.

4.4. ОПИСАНИЕ ПЕРЕМЕННЫХ. ТИПЫ ДАННЫХ

Переменная — это область оперативной памяти, занимающая несколько ячеек и имеющая свое имя. Переменная обладает следующими свойствами:

- переменная хранит не более 1 значения;
- переменная способна хранить значения только одного и того же типа;
- переменная хранит значение до тех пор, пока в нее не поместят новое значение, при этом предыдущее содержимое переменной стирается;
- значение переменной может быть вызвано для использования сколько угодно раз без изменения оригинала;
- к началу выполнения программы содержимое переменной считается неопределенным;

- ячейки памяти, отведенные под переменную путем ее описания, заполняются значениями в ходе выполнения программы с помощью оператора присваивания или процедуры ввода с клавиатуры; этим переменная отличается от константы, значение которой присваивается до выполнения основной программы, в разделе определения констант;
- в Турбо-Паскале имя переменной должно начинаться с латинской буквы, состоять из латинских букв, цифр или знака подчеркивания “_”. Имя должно состоять не более чем из 63 символов (больше не воспринимает компилятор). В имени переменной нельзя использовать пробел.

Различные типы данных занимают в оперативной памяти компьютера разное количество ячеек (емкость одной ячейки — 1 байт). Перед началом вычислений следует сообщить компьютеру, сколько ячеек памяти надо зарезервировать под ту или иную переменную. Для этого в блоке «VAR» раздела описаний программы должны быть описаны все используемые переменные, то есть должно быть указано имя каждой переменной и типы данных, которые будут храниться в этих переменных.

В Турбо-Паскале приняты следующие простые типы данных (типы значений констант и переменных):

Integer (целочисленная переменная)

Для хранения значения в переменной типа `integer` в оперативной памяти компьютера отводится 2 байта памяти. Это может быть целое число в диапазоне от -32768 до $+32767$.

Real (переменная вещественного типа)

Под значение переменной типа `REAL` в памяти отводится 6 байт. Целая часть отделяется от дробной точкой. Переменная типа `real` — это вещественное число в диапазоне от $2,9E-39$ до $1,7E+38$ по модулю. Буква «E» в этой записи означает «умножить на 10 в степени».

Char — это символьный (литерный) тип, в памяти занимает 1 байт, диапазон значений — один символ (одна буква, цифра, знак препинания и т.п.).

String — строковый тип, производный от типа Char. STRING — это строка символов. В памяти строка занимает MAX+1 байт, где MAX — объявленное максимальное количество символов в строке. Например, если некоторая переменная была описана как переменная типа STRING[80], то в нее можно записать не более 80 символов, и в памяти она займет 81 байт.

Если при описании строковой переменной максимальная длина строки точно не указана (слово STRING не сопровождается целым числом), то переменной автоматически дается максимальная для типа STRING размерность — 255 символов, что потребует 256 байт памяти для хранения значения.

Boolean — это логический тип данных, занимает в памяти 1 байт. Переменная этого типа может хранить только одно из двух возможных значений — True (истина) или False (ложь).

4.5. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ В ПАСКАЛЕ

Арифметические операции в Паскале с переменными типа INTEGER и REAL напоминают привычные нам алгебраические или тригонометрические формулы. Отличие в том, что дроби записываются в одну строку и используется необычное обозначение операций умножения (*) и деления (/ , div). При записи формул (выражений) в программе следует учитывать такую особенность: компьютер будет в первую очередь выполнять вычисления в скобках, затем операции умножения и деления и в последнюю очередь — сложение и вычитание:

1. ()
2. *, /
3. +, -

Операции с равным приоритетом (+ , -), (*, /) выполняются слева направо в том порядке, как записаны в выражении. Если у программиста нет уверенности в том, что вычисления будут выполняться в нужном порядке, то лучше расставить лишние скобки, помня о том, что

вычисления в скобках выполняются в первую очередь. Например, в выражении $A+B*C/D-E$, согласно правилам языка Паскаль, сначала будет выполнено $B*C$, затем результат делится на D , и только после этого наступит очередь сложения и вычитания. Если сложение и вычитание требуется выполнить в первую очередь, то следует соответствующим образом расставить скобки: $(A+B)*C/(D-E)$.

Заметим, что *при использовании операции вещественного деления (/) результатом будет вещественное число* (имеющее целую и дробную части). Такое число можно записать только в переменную вещественного типа (типа REAL, способную хранить 6 байт). Попытка записать результат вещественного деления в переменную целого типа (типа INTEGER) будет ошибочной, так как под переменную такого типа в памяти отводится 2 байта, что недостаточно для размещения 6 байт вещественного числа. Следует подчеркнуть, что при использовании вещественного деления результатом будет вещественное число даже в том случае, если операндами являются целые числа, например: $4/2$ получим 2,0 (в числе “2,0” присутствует дробная часть). Для того чтобы результатом деления целых чисел было целое число (способное поместиться в переменную типа INTEGER), следует использовать операцию деления «DIV». Целочисленное деление (div) отличается от обычной операции деления тем, что возвращает целую часть частного, а дробную часть отбрасывает; перед выполнением операции оба операнда округляются до целых значений. Если делимое меньше делителя, то результат целочисленного деления всегда равен нулю.

Выражение	Результат
11 div 5	2
10 div 3	3
2 div 3	0

Помимо операции целочисленного деления, существует и операция, позволяющая определить остаток, получаемый в результате целочисленного деления (mod).

Выражение	Результат
$5 \bmod 2$	1
$11 \bmod 13$	11
$6 \bmod 3$	0

Пример 2. Дано трехзначное число Z . Найти S – сумму цифр данного числа.

Решение. Для решения задачи требуется:

- ввести с клавиатуры трехзначное число Z ;
- отделить цифры числа: последняя цифра: $Z \bmod 10$, первая цифра: $Z \operatorname{div} 100$, средняя цифра: $(Z \operatorname{div} 10) \bmod 10$;
- найти сумму полученных цифр;
- вывести результат на экран.

Текст программы следующий.

```

program P2;
var Z, C1, C2, C3, S : integer;
begin
writeln('введите радиус окружности');
readln(Z);
C1:=Z div 100;
C2:=(Z div 10) mod 10;
C3:=Z mod 10;
S:=C1+C2+C3;
writeln('S=',S);
end.

```

К переменным целого и вещественного типа применимы следующие стандартные функции Паскаля:

Обращение	Тип аргумента	Тип результата	Функция
$\operatorname{abs}(x)$	I, R	I, R	Модуль аргумента
$\operatorname{arctan}(x)$	I, R	R	Арктангенс аргумента (результат в радианах)
$\operatorname{cos}(x)$	I, R	R	Косинус (угол в радианах)
$\operatorname{exp}(x)$	I, R	R	Экспонента e^x

Обращение	Тип аргумента	Тип результата	Функция
$\text{frac}(x)$	I, R	R	Дробная часть аргумента
$\text{int}(x)$	I, R	R	Целая часть аргумента. Если $x \geq 0$, функция возвращает ближайшее целое число меньше или равное x . Если $x < 0$, то результат есть целое число больше или равное x
$\text{ln}(x)$	I, R	R	Вычисление натурального логарифма x , т.е. логарифма по основанию e ($e = 2,718282$)
Pi		R	Возвращает значение числа Пи (3,141592653897932385)
Random		R	Генерирует значение псевдослучайного числа из диапазона [0...1)
randomize			Инициализация генератора псевдослучайных чисел (обновление базы случайных чисел)
$\text{round}(x)$	R	I	Возвращает значение x , округленное до ближайшего целого числа
$\text{sin}(x)$	I, R	R	Синус (угол в радианах)
$\text{sqr}(x)$	I, R	I, R	Возведение аргумента в квадрат
$\text{sqrt}(x)$	I, R	R	Вычисление квадратного корня из аргумента
$\text{trunc}(x)$	R	I	Возвращает ближайшее целое число, меньше или равное x , если $x \geq 0$, и больше или равное x , если $x < 0$

Обращение	Тип аргумента	Тип результата	Функция
odd(x)	I	Boolean	Проверка четности числа. Возвращает значение True, если аргумент — нечетное число, и False, если x — четное
random(x)	I	I	Генерирует значение псевдослучайного числа из диапазона [0.. x]

В Паскале *отсутствует функция возведения числа в степень*, а также нет тригонометрических функций tg(), ctg(), arcsin() и arccos(), но они могут быть реализованы сочетанием стандартных функций:

$x^n = \exp(n * \ln(x))$; — результат возведения в степень таким способом имеет вещественный тип (real)

$$\text{tg}(x) = \sin(x) / \cos(x);$$

$$\text{ctg}(x) = \cos(x) / \sin(x);$$

$$\arcsin(x) = \arctan(x / \text{sqrt}(1 - x * x));$$

$$\arccos(X) = \arctan(\text{sqrt}(1 - x * x) / x).$$

Пример 3. Требуется записать на языке Паскаль математические выражения.

Решение.

Математическое выражение	Выражение на Паскале
--------------------------	----------------------

$$x^2 - 7x + 6$$

$$\text{Sqr}(x) - 7 * x + 6$$

$$\frac{|x| - |y|}{1 + |xy|}$$

$$(\text{abs}(x) - \text{abs}(y)) / (1 + \text{abs}(x * y))$$

$$\ln \left(\left(y - \sqrt{|x|} \right) \left(x - \frac{y}{z + x^2} \right) \right) \ln \left(\text{abs} \left(\begin{array}{l} (y - \text{sqrt}(\text{abs}(x))) * \\ * (x - y / (z + \text{sqrt}(x))) \end{array} \right) \right)$$

4.6. ОПЕРАЦИИ ОТНОШЕНИЯ И ЛОГИЧЕСКИЕ ОПЕРАЦИИ

Операции отношения позволяют сравнивать 2 значения, результатом сравнения являются данные логического (*boolean*) типа, а именно — правда или ложь (*true* или *false*).

В Паскале имеются следующие операции отношения:

= равно

<> не равно

< меньше

> больше

<= меньше или равно

>= больше или равно

Если сравнение справедливо, то результатом будет правда (*true*), в противном случае — ложь (*false*).

Построенные с помощью операций отношения простые пары сравнений типа $A > B$ или $C = 3,14$ могут объединяться в более сложные логические выражения с помощью логических операций: OR (или), AND (и), NOT (не), XOR (истинно либо то, либо другое, но не оба вместе).

Логические операции имеют приоритет (старшинство) над операциями отношения и выполняются в первую очередь, поэтому в сложных выражениях операции отношения заключаются в скобки:

$A = B \text{ AND } C > D$ неправильно,

$(A = B) \text{ AND } (C > D)$ правильно.

4.7. ОСНОВНЫЕ ОПЕРАЦИИ ЯЗЫКА ТУРБО-ПАСКАЛЬ

4.7.1. Оператор присваивания

Как известно, переменная — это несколько ячеек оперативной памяти, имеющих общее имя и предназначенных для хранения одного значения определенного типа. До выполнения вычислений значение переменной не определено, ячейки памяти заполнены случайной информаци-

ей, оставшейся от предыдущих программ. Запоминание переменной своего значения, или, иначе говоря, запись в ячейки памяти данных, выполняется с помощью оператора присваивания, например $A:=2$.

В Паскале знак присваивания представлен комбинацией двух символов: “:=” и “=”. Знак присваивания не следует путать с операцией сравнения “=”. В случае ошибочной записи оператора присваивания, например в выражении $i:=i+1$ (переменная i увеличивает свое значение на 1), он будет выглядеть как сравнение $i=i+1$, заведомо ложное.

Выполнение оператора присваивания сводится к вычислению значения выражения, стоящего от символа присваивания справа, с последующей записью полученного результата в переменную, стоящую от символа слева:

```
C:=2*pi*R;
S:=pi*sqr(R).
```

Не допускается записывать в целочисленную переменную (типа Integer) вещественное значение (типа Real).

В логическом операторе присваивания слева от знака присваивания указывается переменная логического типа (boolean), а справа дается логическое выражение, имеющее значение True (правда) или False (ложь):

```
D:=True;
V:=(A>C) AND (D<>0).
```

В символьном и строковом операторе присваивания слева от знака стоит переменная типа Char или String, а справа — символ, строка символов либо символьное или строковое выражение:

```
SYM:='A'; Alpha:=SUM;
Str1:='Turbo-'; Str2:='Pascal'; Str:=Str1+Str2.
```

Пример 4. Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

```
a:= 1819;
b:= (a div 100)*10+9;
a:= (10*b-a) mod 100;
```


$$\begin{aligned} 1) \quad a &= 81, \\ b &= 199 \end{aligned}$$

$$\begin{aligned} 2) \quad a &= 81, \\ b &= 189 \end{aligned}$$

$$\begin{aligned} 3) \quad a &= 71, \\ b &= 199 \end{aligned}$$

$$\begin{aligned} 4) \quad a &= 71, \\ b &= 189 \end{aligned}$$

Решение. Выполним операции присваивания:

$$a = 1819;$$

$$b = 18 \cdot 10 + 9 = 189;$$

$$a = (10 \cdot 189 - 1819) \bmod 100 = (1890 - 1819) \bmod 100 = 71 \bmod 100 = 71.$$

Верный ответ: 4.

4.7.2. Составной оператор

При написании программ на Паскале часто требуется, чтобы в определенном месте несколько операторов условно фигурировали (представлялись) как один оператор. Для этого используется составной оператор, который объединяет последовательность (цепочку) операторов в единый оператор путем обрамления этой последовательности операторными скобками — словами “begin” и “end”. Внутри операторных скобок операторы отделяются друг от друга знаком “;”. Выполнение составного оператора сводится к последовательному (в порядке их написания) выполнению входящих в него операторов:

```
begin
i:=0;
j:=0
end;
```

4.7.3. Операторы ввода-вывода

Операторы ввода-вывода необходимы для ввода в компьютер данных с клавиатуры либо вывода результатов расчета на экран дисплея.

Для **ввода данных** используется оператор “Readln”, например Readln (A), где A — имя переменной.

Встретив в тексте программы оператор Readln (имя переменной), компьютер останавливается и ждет, когда с

клавиатуры будет введено значение для указанной переменной и нажата клавиша ENTER. После этого ячейка оперативной памяти, отведенная для переменной, будет заполнена введенными данными.

Оператор `Readln (A,B,C)` ожидает ввода с клавиатуры трех переменных. Если эти переменные имеют числовой тип, то можно ввести значения переменных в одной строке, разделяя их пробелом, и в конце нажать клавишу ENTER, или можно после ввода каждого числа нажимать клавишу ENTER, тогда ввод займет три строки. Последний способ является единственно верным и при вводе строковых значений.

Заметим, что при вводе значений переменных, так же как и при присваивании, очень важно соответствие типов переменных и их значений. Например, если попытаться ввести в переменную целочисленного типа (`Integer`) вещественное число, то произойдет программное прерывание. Кстати, обратное действие осуществимо: переменной вещественного типа (`Real`) можно присвоить целое значение, в этом случае дробная часть числа будет равна нулю.

Вывод данных на экран осуществляется операторами `Write (A, B, C, ... , Z)`, либо `Writeln (A, B, C, ... , Z)`, где `A, B, C, ... , Z` — имена переменных либо выражения.

Разница между этими двумя схожими операторами в следующем: при выполнении оператора `Writeln ()` все перечисленные в нем элементы печатаются в одну строку в заданном порядке, после чего курсор переводится в начало следующей строки. Если после вывода последнего элемента списка необходимо оставить курсор на той же строке, следует использовать оператор вывода `Write ()`.

Часто при программировании необходимо вывести на экран небольшие текстовые сообщения: заголовок таблицы, пояснение к вводимым или выводимым данным и т.п. Для этого выводимый текст ограничивается с обеих сторон апострофами и записывается в оператор вывода:

```
Writeln('Введите исходные данные');
```

Такой оператор выводит на экран все символы, заключенные между апострофами, но сами апострофы не печатаются.

Если в операторе вывода числовые переменные отделяются друг от друга только запятыми, то при выводе на экран их значения будут сливаться в одну строку, что затруднит чтение. Так, если переменные A, B соответственно имеют значения 3 и 4, то при выполнении оператора `Writeln (A,B);` на экране появится строка

```
34
```

Чтобы выводимые числовые значения не сливались визуально в одно число, перечисляемые в операторе `Writeln ()` переменные следует разделять пробелами, заключенными в апострофы:

```
Writeln(A, ' ', B).
```

Другой способ разделения выводимых данных заключается в использовании формата длины поля:

```
Writeln(A:N),
```

где A — имя переменной; N — число знакомест, резервируемых в строке на экране под выводимое значение переменной.

Если количество цифр в значении переменной окажется меньше, чем было заказано в длине поля вывода (N), то выводимые символы будут выравниваться по правой границе отведенного поля, а в оставшееся слева от числа незаполненное пространство поместятся пробелы. Если длина поля окажется недостаточной для печати числа, Паскаль автоматически увеличит длину поля до минимально необходимого размера.

При выводе на экран значений вещественных переменных (типа Real) в формате поля вывода необходимо, кроме длины поля, указывать количество знакомест, резервируемых под дробную часть числа. В противном случае число будет распечатано в неудобной для чтения экспоненциальной форме.

```
Writeln(A:N:K),
```

здесь *A* — имя переменной типа “Real”; *N* — общая длина поля; *K* — число знакомест в длине поля, отводимых под дробную часть числа.

Например:

```
X:=248.74; Writeln(X:6:2).
```

Пример 5. Требуется написать программу возведения числа *A* в степень *N*.

```
Program Stepen;
Var A,B,N: Real;
Begin
Writeln('Программа возведения числа в степе-
пень');
Write('Введите число = '); Readln(A);
Write('Укажите степень = '); Readln(N);
B:=Exp(N*Ln(A));
Writeln('Результат:');
Writeln('B= ',B:8:2);
Readln
End.
```

Если результаты расчета надо вывести не на экран, а на принтер, то в раздел описаний программы надо добавить блок *Uses*, в котором указать имя модуля *PRINTER*. Далее, в тех операторах вывода (*Write*, *Writeln*), которые должны выводить данные на бумагу, надо добавить после открывающейся скобки слово *LST*, например:

```
Writeln (LST,'B= ',B:8:2).
```

4.7.4. Условный оператор

4.7.4.1. Понятие об условном операторе

Программы, составленные исключительно из операторов присваивания и ввода-вывода, могут иметь только линейную структуру.

Между тем часто бывает нужно, чтобы какой-нибудь блок программы выполнялся или не выполнялся в зависимости от задаваемого условия. Для подобного разветвления вычислительного процесса служат условные опе-

раторы. В Паскале имеется 2 формы условного оператора: полная и сокращенная.

4.7.4.2. Полный условный оператор

В общем виде полный условный оператор выглядит следующим образом:

```
IF <условие> THEN Оператор1 ELSE Оператор2;
```

Эта запись читается следующим образом: “если логическое выражение справедливо, то выполняется первый оператор, если ложно — второй оператор”. Операторы 1 и 2 могут быть простые или составные. Точка с запятой перед словом ELSE не ставится.

Примеры полных условных операторов:

```
IF x < 0 THEN i:=i+1 ELSE j:=j+1;
IF (x < y) AND (x < > 0) THEN begin x:=y/x;
y:= y+k end
ELSE begin x:=x+0.01; y:=y-k end;
```

Пример 6. Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования).

```
a := 200;
b := 50;
a := a + b * 3;
if a > b then c := a - b
else c := b - a;
```

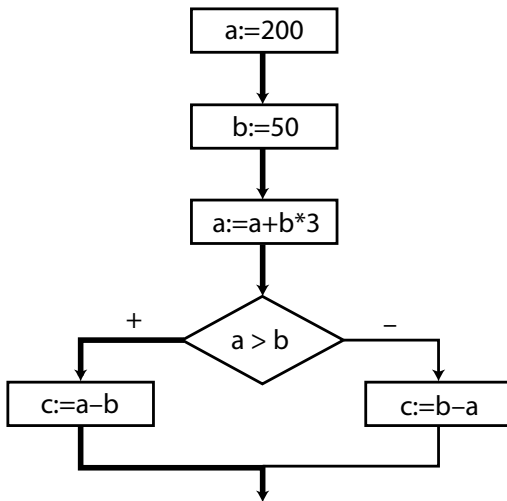
Для верного решения данного задания предлагаем вам записать блок-схему параллельно с вычислениями:

```
{a = 200 + 50 * 3 = 200 + 150 = 350}
```

```
{350 > 50}
```

```
{c = 350 - 50 = 300}
```

Обратите внимание на порядок выполнения арифметических действий (см. пункт 4.5). Жирной линией показано исполнение алгоритма.



Верный ответ: 300.

4.7.4.3. Сокращенный условный оператор

Сокращенный условный оператор можно записать так:

```
IF <условие> THEN <Оператор>;
```

Эта запись читается следующим образом: «если логическое выражение справедливо, то выполняется указанный оператор, в противном случае никаких дополнительных изменений в вычисления не приносится».

Примеры использования условного оператора в сокращенной форме:

```
IF x>y THEN Max:=x;
IF x<>0 THEN
begin
a:=y/x;
writeln('A= ', a:6:2)
end;
```

4.7.4.4. Создание сложных разветвлений алгоритма

Операторы, включаемые в условный оператор, в свою очередь также могут быть вложенными.

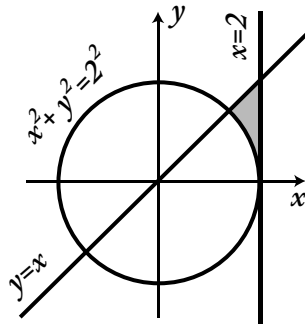
Использование таких условных операторов, «вложенных» один в другой, требует осторожности, чтобы не запутаться в их структуре. Например, условный оператор

```
IF <условие 1> THEN IF <условие 2> THEN
<оператор 1> ELSE <оператор 2>;
```

можно истолковать двояко:

```
1) IF <условие 1> THEN
begin
IF <условие 2> THEN <оператор 1>
end
ELSE <оператор 2>;
```

```
2) IF <условие 1> THEN
begin
IF <условие 2> THEN <оператор 1>
ELSE <оператор 2>
end;
```



По правилам языка Паскаль считается, что каждое слово ELSE соответствует первому предшествующему ему слову THEN. Во избежание ошибок рекомендуется четко выделять желаемую форму условного оператора с помощью операторных скобок begin ... end.

Пример 7. 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.

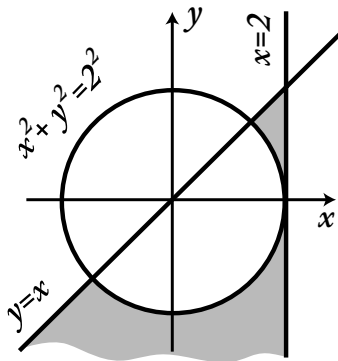
```
var x,y:real;
begin
  readln(x,y);
  if x*x+y*y >=4 then
    if x >=2 then
      if y >=x then write('принадлежит')
    else write('не принадлежит')
  end.
```

Выполните последовательно действия:

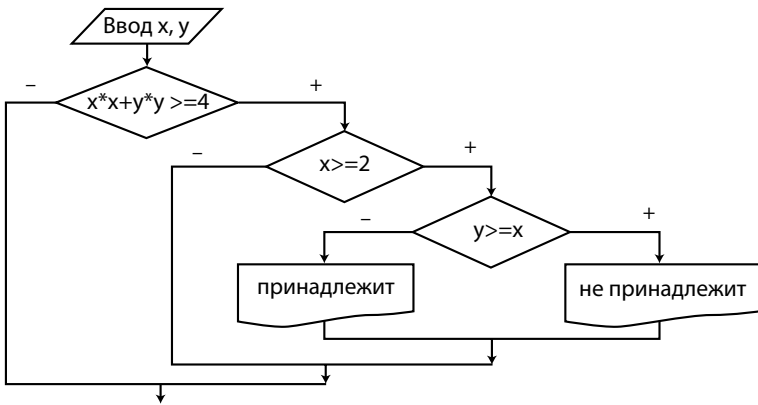
1. Приведите пример таких чисел x , y , при которых программа неверно решает поставленную задачу.

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

Программист допустил две ошибки, для того чтобы их обнаружить, построим блок-схему и нарисуем область, описанную программистом.



Строим блок-схему.



Из блок-схемы явно видно, что не во всех случаях программа выдает тот или иной ответ. Поэтому в качестве примера чисел x , y , при которых программа неверно решает поставленную задачу, можно взять значения, не удовлетворяющие первому или второму условию. Например, $x = 0$, $y = 0$, или $x = 5$, $y = 2$. Кроме того, заметим, что программист неверно описал заштрихованную область. В его исполнении получаем следующую область. Поэтому в качестве примера чисел x , y , при которых программа неверно решает поставленную задачу, можно также взять значения из той области, которая заштрихована ниже окружности, например $x = 1$, $y = -3$.

Для решения второй части задачи необходимо исправить обе ошибки. Во-первых, нужно добавить еще одно условие $y \geq 0$ и, во-вторых, добавить недостающий вывод информации:

```

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

```
else write('не принадлежит')
else write('не принадлежит')
else write('не принадлежит')
end.
```

Предложенный вариант является далеко не единственным и не самым изящным. Предложим вам более короткий вариант:

```
var x,y:real;
begin
  readln(x,y);
  if (x*x+y*y>=4) and (x>=2) and (y>=x) and (y>=0)
  then write('принадлежит')
  else write('не принадлежит')
end.
```

Заметим, что при записи составных условий каждое элементарное условие записывается в круглых скобках.

Пример 8. Составить программу, которая при любых значениях “а” и “b” решает неравенство $ax + b \leq 0$.

Решение. С математической точки зрения данная задача является неравенством с параметром и имеет следующее решение:

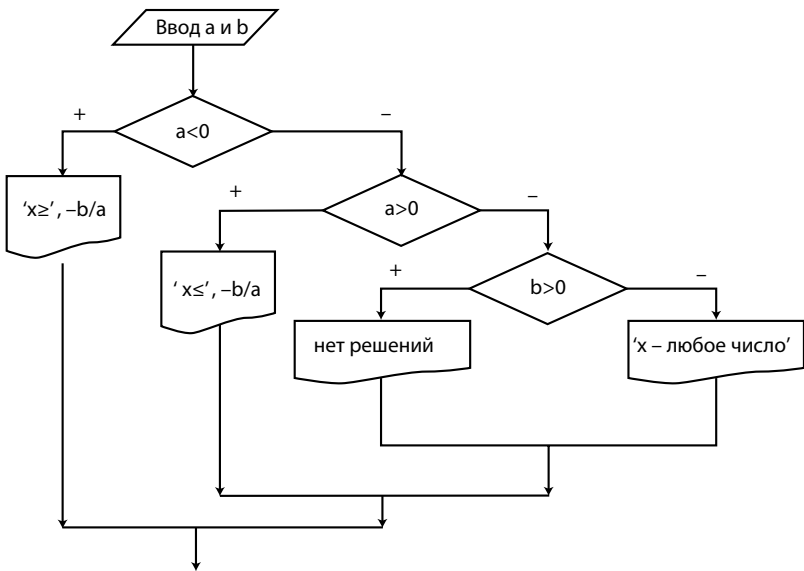
$$x \leq -\frac{b}{a}, \text{ при } a > 0,$$

$$x \geq -\frac{b}{a}, \text{ при } a < 0,$$

$$x \in R, \text{ при } a = 0 \text{ и } b \leq 0,$$

нет решений при $a = 0$ и $b > 0$.

Видим, что определяющую роль при решении неравенства имеет параметр “а”, и лишь в том случае, если он равен нулю, анализируется значение параметра “b”. Построим блок-схему.



На основе блок-схемы составим программу.

```

var a,b:real;
begin
writeln('введите a и b')
readln(a,b);
if a<0 then writeln('x>=',-b/a)
else if a>0 then writeln('x<=',-b/a)
else if b>0 then writeln('нет решений')
else writeln('x - любое число');
readln;
end.
  
```

4.7.5. Оператор варианта «CASE»

Характерной чертой некоторых алгоритмов является широкая разветвленность задаваемого ими вычислительного процесса. Использование в таких случаях условного оператора IF ... THEN может оказаться громоздким и недостаточно наглядным, поэтому лучше применять оператор варианта. Общий вид оператора варианта таков:

```

CASE селектор OF
Список 1: оператор1;
Список 2: оператор2;
.....
Список N: операторN
ELSE альтернативный оператор
END;

```

Оператор CASE состоит из селектора и списка операторов, каждому из которых предшествует список вариантов, представляющий собой значение того же типа, что и селектор. Селектор должен относиться к одному из целочисленных (находящихся в диапазоне $-32768 \dots 32767$), логическому или литерному типов. Вещественные и строковые типы в качестве селектора запрещены. Список констант выбора состоит из произвольного количества значений или диапазонов, отделенных друг от друга запятыми. Границы диапазона записываются двумя константами через разграничитель «...». Тип констант в любом случае должен совпадать с типом селектора.

При работе оператора CASE будет выполнен тот входящий в него оператор, чей список содержит текущее значение селектора. Если ни один из списков не содержит значения селектора, то не выполняется ни один оператор. Следует оговориться, что Турбо-Паскаль, в отличие от стандартного Паскаля, допускает использование в операторе CASE ключевого слова ELSE. В случае отсутствия значений в списках, равных значению селектора, выполняется оператор, стоящий за словом ELSE (альтернативный оператор).

Пример 9. Дано целое число N. Требуется записать фразу «я нашел N гриб(-ов, -а)».

Решение. Окончание существительного «гриб» зависит от числительного N:

N	окончание
1, 21, 31, 41, 51, 61, ...	–
2..4, 22..24, 32..34, 42..44, ...	-а
5..10, 11..20, 25..30, 35..40, ...	-ов

Заметим, что в общую схему не укладывается диапазон 11..20, в остальных случаях все зависит от последней цифры числа N.

```
Program Z9;
var N,K:integer;
ok:string[2];
begin
writeln('введите число N');
readln(N);
if (N>=10) and (N<=20) then ok:='ов'
else begin
K:=N mod 10;
case K of
1: ok:='';
2..4: ok:='a';
5..9,0: ok:='ов'
end
end;
writeln('Я нашел ',N,' гриб',ok);
readln
end.
```

4.7.6. Комментарии

При программировании желательно вставлять в текст создаваемой программы **КОММЕНТАРИЙ** — несколько фраз, поясняющих суть выполняемых программой действий. В ходе трансляции текста программы компилятор обходит комментарии, не принимает их во внимание и не включает в исполняемые коды программы. Единственное назначение комментария — служить справочной информацией для человека, читающего текст программы. Отсутствуют универсальные правила для определения мест, где следует располагать комментарии. Размещение и содержание комментариев в основном зависит от личных представлений программиста. Грамотно представленные комментарии позволяют ответить на большинство вопросов, возникающих при знакомстве с текстом программы:

что делает данная программа, что обозначает каждая переменная, какова цель данного оператора и т.д. В Паскале комментарии создаются обрамлением с обеих сторон какой-нибудь текстовой строки условными символами { } или (* *), например:

```
{В этом месте программа ожидает ввода числа}  
или
```

```
(* Конец раздела описаний, начинаем основную  
программу *).
```

4.7.7. Оператор цикла

4.7.7.1. Понятие о цикле

При разработке алгоритмов решения большинства задач возникает необходимость многократного повторения однотипных команд. Это реализуется с помощью операторов цикла. В Паскале имеется три разновидности оператора цикла: цикл с параметром, цикл с предусловием и цикл с постусловием.

4.7.7.2. Оператор цикла с параметром

Общий вид оператора цикла с параметром:

```
FOR i:= A TO B DO <тело цикла>;
```

где i — параметр цикла (счетчик повторов), переменная целого или символьного типа (Integer, Char);

A и B — начальное и конечное значения параметра цикла — выражения того же типа, что и параметр цикла;

тело цикла — любой простой или составной оператор, который требуется повторить несколько раз.

Оператор цикла типа FOR ... TO ... DO предусматривает последовательное **увеличение на единицу** параметра цикла « i » от начального значения «A» до конечного значения «B» и выполнение тела цикла при каждом значении параметра цикла.

Оператор цикла типа FOR ... DOWNTO ... DO предусматривает последовательное **уменьшение на единицу** параметра цикла « i » от начального значения «A» до конечного

значения «В» и выполнение тела цикла при каждом значении параметра цикла.

Для операторов цикла с параметром существуют некоторые ограничения:

- нельзя задавать шаг изменения значения параметра, отличный от 1 или -1;
- не изменять внутри цикла значения параметра цикла, начальное и конечное значения параметра;
- после выхода из цикла значение параметра не определено;
- входить в цикл можно только через его начало, а выходить — либо при исчерпании значений параметра цикла, либо при выполнении оператора прерывания цикла **BREAK**.

Пример 10. Найти сумму 10 чисел, введенных с клавиатуры.

Решение. В переменную *X* 10 раз введем с клавиатуры число, а к сумматору *S* 10 раз добавим *X*. Для того чтобы 10 раз произвести эти действия, используем цикл с параметром.

```
Program Z10;
var X,S,I : integer;
begin
S:=0;
for I:=1 to 10 do
begin
writeln('введите число X');
readln(X); {вводим число X}
S:=S+X; {прибавляем X к сумме}
end;
writeln('S=',S);
readln
end.
```

Так как тело цикла содержит 3 оператора, то необходимо заключить их в операторные скобки `begin ... end`.

Исполнение алгоритма

Значение параметра I	Значение переменной X (могут быть любыми другими)	Значение переменной S
1	5	$0 + 5 = 5$
2	6	$5 + 6 = 11$
3	1	$11 + 1 = 12$
4	3	$12 + 3 = 15$
5	-5	$15 + (-5) = 10$
6	9	$10 + 9 = 19$
7	8	$19 + 8 = 27$
8	2	$27 + 2 = 31$
9	-4	$31 - 4 = 27$
10	7	$27 + 7 = 34$

Пример 11. Дано натуральное число N. Вычислить N! (см. п. 1.4).

Решение. Переменную F, изначально равную 1, будем последовательно домножать на 1, 2, 3, ..., N, то есть в качестве множителя можно использовать параметр цикла.

```

Program Z11;
var N,F,I : integer;
begin
write('введите N=');
readln(N);
F:=1;
for I:=1 to N do
F:=F*I;
writeln(N,'!=',F);
readln
end.

```

Исполнение алгоритма при N=5.

Значение параметра I	Значение переменной F
1	$1*1 = 1$
2	$1*2 = 2$
3	$2*3 = 6$
4	$6*4 = 24$
5	$24*5 = 120$

Пример 12. Дан код программы. Какое значение переменной “с” было дано на входе программы, если на выходе получилось значение $s = 5050$?

```
var c, i, s : integer;
begin
  readln(c);
  s:=0;
  for i:=1 to c do
    s:=s+c;
  writeln(s);
end.
```

Решение. В переменную s поочередно складываются значения $1, 2, \dots, c$, то есть $s = 1 + 2 + 3 + \dots + c$. Из курса математики следует, что s есть сумма c первых членов арифметической прогрессии, в которой $a_1 = 1, a_c = c$, значит, $s = \frac{a_1 + a_c}{2} \cdot c = \frac{1 + c}{2} \cdot c = 5050$. В итоге получаем квадратное уравнение $c^2 + c - 10100 = 0$, тогда $c = 100$.

Верный ответ: 100.

4.7.7.3. Оператор цикла с предусловием

Если число повторений, выполняемых в цикле, заранее неизвестно или шаг приращения счетчика (параметра) цикла отличен от единицы, то необходимо использовать оператор цикла с предусловием. Оператор цикла этого вида имеет вид:

```
WHILE <условие> DO <тело цикла>;
```

где условие — это логическое выражение, от значения которого зависит — продолжать повторы или завершить цикл;

тело цикла — любой простой или составной оператор.

Выполнение оператора начинается с вычисления значения логического выражения. Если оно имеет значение «True» (истина), то выполняется тело цикла. Выполнение цикла продолжается до тех пор, пока логическое выражение в его заголовке не примет значение «False» (ложно). Если выражение равно «False» при первом же витке цик-

ла, то работа цикла завершится, а входящие в него операторы не выполнятся ни разу. Поскольку в цикле типа WHILE ... DO условие завершения его работы проверяется до выполнения входящего в него оператора, такой цикл называется “оператор цикла с предусловием”.

Пример 13. Дано натуральное число N. Определить:

- количество цифр в записи числа N;
- сумму цифр в записи числа N;
- первую цифру числа N.

Решение. Для решения этой задачи используем операции целочисленного деления и остатка от целочисленного деления. Число делится на цифры, начиная с наименьшего разряда, в результате деления на 10. Деление продолжается до тех пор, пока от числа не останется одна цифра.

```

Program Z13;
var N,D,K,S,P : integer;
begin
write('введите N=,');
readln(N);
K:=0; {количество цифр}
S:=0; {сумма цифр}
D:=N; {дубликат числа N}
While D>=10 do {пока в числе более одной цифры}
Begin
P:=D mod 10; {последняя цифра числа}
K:=K+1; {увеличиваем количество цифр в числе}
S:=S+P; {прибавляем к сумме последнее число}
D:=D div 10; {убираем последнюю цифру в числе}
End;
K:=K+1;
S:=S+D;
writeln('количество цифр в числе: ',K);
writeln('сумма цифр в числе: ',S);
writeln('первая цифра в числе: ',D);
readln
end.

```

Исполнение алгоритма при $N = 14023$. $D = 14023$

Проверка условия $D \geq 10$	Значение переменной P	Значение переменной K	Значение переменной S	Значение переменной D
$14023 \geq 10$	3	$0 + 1 = 1$	$0 + 3 = 3$	1402
$1402 \geq 10$	2	$1 + 1 = 2$	$3 + 2 = 5$	140
$140 \geq 10$	0	$2 + 1 = 3$	$5 + 0 = 5$	14
$14 \geq 10$	4	$3 + 1 = 4$	$5 + 4 = 9$	1
$1 < 10$	Цикл завершен, переходим к следующему после цикла оператору			
		$4 + 1 = 5$	$9 + 1 = 10$	

Пример 14. После выполнения фрагмента программы переменная S приняла значение 4. Определить минимальное целое N , при котором это возможно.

```

S:=0;
i:=1;
while i<N do
begin
if i mod 3=0 then S:=S+1;
i:=i+1;
end;
S:=S*S-2*S+1;

```

Решение. Начнем рассуждения с конца программы. Окончательное значение переменной S получается из предыдущего после присваивания $S := S * S - 2 * S + 1$. Значит, предпоследнее значение S (после выхода из цикла) получаем из уравнения $S * S - 2 * S + 1 = 4$, или $(S - 1)^2 = 4$, то есть $S = 3$ или $S = -1$. Так как в теле цикла значение переменной S может только увеличиваться, начиная с 0, то после выхода из цикла $S = 3$. Теперь проанализируем тело цикла

```

if i mod 3=0 then S:=S+1;
i:=i+1;

```

Значение переменной S увеличивается на 1 в том случае, если i кратно 3. При $i = 3$ получаем $S = 1$, при $i = 6$: $S = 2$ и при $i = 9$: $S = 3$, таким образом $i = 9$ достаточно, однако в

этом же проходе тела цикла i увеличивается на 1, то есть $i=10$. Так как условие выполнения тела цикла $i < N$, то для выхода из цикла должно выполняться условие $i \geq N$, или $10 \geq N$, то есть $N=10$.

Верный ответ: 10.

4.7.7.4. Оператор цикла с постусловием

Цикл этой разновидности применяется в случаях, когда число повторений оператора, входящего в тело цикла, заранее неизвестно. Такой цикл похож на цикл с предусловием, но в данном случае условие завершения повторов проверяется после выполнения операторов, составляющих тело цикла. Общий вид оператора цикла с постусловием таков:

```
REPEAT
<оператор1>, <оператор2>, ... , <операторN>
UNTIL <условие>;
```

где оператор1, оператор2, ... , операторN — операторы тела цикла;
условие — логическое выражение, диктующее завершение повторов.

Оператор цикла с постусловием начинается с выполнения операторов внутри цикла. Затем проверяется истинность логического условия, стоящего после слова UNTIL. Если это условие справедливо (True), то осуществляется выход из цикла. Если же значение логического выражения ложно (False), то выполнение операторов тела цикла повторяется, после чего снова проверяется истинность логического условия. Заметим, что в цикле с постусловием тело цикла выполняется хотя бы один раз.

Очень часто цикл с постусловием используется для проверки вводимой информации.

Пример 15. Дано натуральное число N . Вычислить

$$n!! = \begin{cases} 1 \cdot 3 \cdot 5 \cdot \dots \cdot n, & n - \text{четное,} \\ 2 \cdot 4 \cdot 6 \cdot \dots \cdot n, & n - \text{нечетное.} \end{cases}$$

Решение. При составлении программы учтем проверку вводимой информации с помощью оператора цикла с постусловием:

```

.....
repeat
write('введите натуральное число: ');
readln(n);
until n>0;
.....

```

До тех пор пока $n \leq 0$, будут повторяться операторы, входящие в тело цикла. Для вычисления двойного факториала будем последовательно домножать переменную F на числа $i, i-2, i-4$ и т.д. до тех пор, пока значение переменной i не станет меньше 2 (то есть $i=0$ или $i=1$). Выполнение повторяющихся действий осуществим с помощью оператора цикла с постусловием.

```

Program P15;
var i,n,F:integer;
begin
repeat
write('введите натуральное число: ');
readln(n);
until n>0;
F:=1;
i:=n;
repeat
F:=F*i;
i:=i-2;
until i<=1;
writeln(n,'!!=',F);
readln;
end.

```

4.7.7.5. Правила использования операторов цикла

При использовании операторов цикла следует учитывать следующие особенности.

Операторы, входящие в цикл Repeat ... Until, всегда выполняются хотя бы один раз, поскольку истинность

логического выражения в цикле этого типа проверяется после операторов, входящих в тело цикла. При использовании цикла типа WHILE ... DO могут быть ситуации, когда операторы, входящие в цикл, не будут выполнены ни разу, если логическое выражение изначально имеет значение "False".

Цикл Repeat ... Until выполняется, пока логическое выражение имеет значение "False". Цикл While ... Do выполняется, пока логическое выражение имеет значение "True".

Если тело цикла "WHILE ... DO" состоит из нескольких операторов, их следует обрамлять операторными скобками "begin ... end", образующими составной оператор. В цикле типа "REPEAT ... UNTIL" операторные скобки не нужны.

Одной из опасностей, возникающих при использовании оператора цикла, является «зацикливание», то есть возникновение ситуации, когда программа не выходит из цикла, бесконечно повторяя входящие в него операторы:

```
k1:=1; WHILE k1<5 DO Writeln('Бесконечный
цикл');
```

Если в программе встретится такой фрагмент, то экран заполнится строками с фразой "Бесконечный цикл", выдача которой в режиме прокрутки (Scrolling) будет длиться до тех пор, пока программистом не будут приняты действия по принудительному завершению работы программы. Вывести программу из бесконечного цикла можно, дважды нажав сочетание клавиш "Ctrl" и "Break", после чего надо нажать клавишу "ESC".

Зацикливания не произойдет, если операторы, входящие в тело цикла, будут влиять на условие, определяющее завершение цикла:

```
k1:=1;
WHILE k1<5 DO
Begin
Writeln('Цикл');
k1:=k1+1
end;
```

В этом фрагменте цикл выполнится ровно 4 раза, и компьютер перейдет к следующим командам.

Циклы могут быть вложены друг в друга, заметим, что в случае использования вложенных циклов с параметром параметрами циклов должны быть разные переменные.

Пример 16. Натуральное число M называется совершенным, если оно равно сумме всех своих делителей, включая единицу, но исключая себя (например, $6=1+2+3$). Требуется напечатать все совершенные числа, меньшие заданного числа N .

Решение.

В начале программы проверим правильность вводимой информации:

```
repeat
write('введите натуральное N=');
readln(N);
until N>0;
```

Затем введем счетчик количества найденных совершенных чисел:

```
k:=0;
```

Далее организуем цикл, в котором будем находить сумму множителей всех чисел в диапазоне от 2 до $N - 1$:

```
for i:=2 to N-1 do
```

В этом цикле для каждого числа i будем проверять, делится ли оно на числа в промежутке от 2 до $i/2$ (при использовании цикла с параметром деление должно быть целочисленным), если да, то будем суммировать делители:

```
S:=1;
for j:=2 to i div 2 do
if i mod j=0 then S:=S+j;
```

Затем будем проверять, совпадает ли само число с суммой своих делителей, если да, то печатаем число и увеличиваем счетчик количества совершенных чисел на 1:

```
if i=S then
begin
k:=k+1;
writeln(i);
end
```

И в завершение программы, после того как проверены все числа из диапазона от 2 до $N - 1$, оцениваем счетчик совершенных чисел, если он равен нулю, то сообщаем о том, что совершенные числа не найдены.

```
if k=0 then writeln('в данном диапазоне нет
совершенных чисел');
```

Полностью программа имеет следующий вид:

```
Program N16;
var N,i,j,k,S:integer;
begin
repeat
write('введите натуральное N=');
readln(N);
until N>0;
k:=0;
for i:=2 to N-1 do
begin
S:=1;
for j:=2 to i div 2 do
if i mod j=0 then S:=S+j;
if i=S then
begin
k:=k+1;
writeln(i);
end
end;
if k=0 then writeln('в данном диапазоне нет
совершенных чисел');
readln
end.
```


4.8. МАССИВЫ

4.8.1. Понятие о массиве

При решении задач с использованием большого количества однотипных упорядоченных данных часто возникает необходимость объединять ряд переменных, хранящих значения одинакового типа, в единую структуру, имеющую общее имя. Для этих целей существуют массивы.

Массив — упорядоченная группа фиксированного количества переменных одного типа, имеющая общее имя.

«Группа переменных» в данном смысле означает, что каждый элемент массива является переменной, для хранения значения которой в памяти отводится определенная область.

«Упорядоченная» — означает, что переменные в массиве хранятся не хаотично, а в установленном порядке, по очереди.

«Фиксированное количество» — означает, что количество переменных в массиве не бесконечно, оно ограничено определенным числом (размерностью массива), которое указывается при описании массива.

«Одного типа» — означает, что переменные, входящие в массив, должны хранить данные одного типа (*Real*, *Integer*, *Char*, *Boolean* и т.д.), который называется типом элементов массива.

Каждому массиву дается собственное имя (идентификатор). По аналогии с переменными, значения элементов массива (переменных, составляющих массив) до выполнения программы не определены. Значения в элементы массива записываются с помощью оператора присваивания.

Над переменными, составляющими массив (над элементами массива), можно выполнять те же действия, что и над обычными переменными — присваивать и считывать значения, преобразовывать значения и сравнивать их.

Каждый элемент массива имеет свое имя, представляющее собой имя массива и стоящие за ним квадратные скобки, в которых указывается индекс элемента — целое число, равное порядковому номеру этого элемента в мас-

сиве. Таким образом, для ссылки на отдельный элемент массива используется запись вида: имя массива[индекс].

Например: A[1], D[24], Massiv[5], Mas[12].

Обработка массива осуществляется в большинстве случаев поэлементно с использованием циклических операторов.

4.8.2 Одномерные массивы

4.8.2.1. Структура линейного массива

Чтобы наглядно представить себе структуру данных в виде массива, рассмотрим пример.

Пример 17. В программе используется одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент программы, в котором значения элементов сначала задаются, а затем меняются.

```
for i:=1 to 10 do
  A[i]:=i*2;
for i:=1 to 10 do
begin
  A[10-i+1]:=A[i];
  A[i]:=A[10-i+1];
end;
```

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 20 18 16 14 12 10 8 6 4 2
- 2) 2 4 6 8 10 10 8 6 4 2
- 3) 0 2 4 6 8 10 12 14 16 18
- 4) 20 18 16 14 12 10 10 12 14 16 18 20

Решение. Разберем выполнение этой программы по частям.

1. Задание значений элементов массива с помощью оператора присваивания.

```
for i:=1 to 10 do
  A[i]:=i*2;
```

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	12	14	16	18	20

2) Изменение элементов массива.

```

for i:=1 to 10 do
begin
A[10-i+1]:=A[i];
A[i]:=A[10-i+1];
end;

```

Пошагово блок 2 выполняется следующим образом:

$i = 1$, тогда $A[10] = A[1] = 2$, $A[1] = A[10] = 2$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	12	14	16	18	2

$i = 2$, тогда $A[9] = A[2] = 4$, $A[2] = A[9] = 4$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	12	14	16	4	2

$i = 3$, тогда $A[8] = A[3] = 6$, $A[3] = A[8] = 6$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	12	14	6	4	2

$i = 4$, тогда $A[7] = A[4] = 8$, $A[4] = A[7] = 8$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	12	8	6	4	2

$i = 5$, тогда $A[6] = A[5] = 10$, $A[5] = A[6] = 10$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	10	8	6	4	2

$i = 6$, тогда $A[5] = A[6] = 10$, $A[6] = A[5] = 10$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	10	8	6	4	2

$i = 7$, тогда $A[4] = A[7] = 8$, $A[7] = A[4] = 8$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	10	8	6	4	2

$i = 8$, тогда $A[3] = A[8] = 6$, $A[8] = A[3] = 6$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	10	8	6	4	2

$i = 9$, тогда $A[2] = A[9] = 4$, $A[9] = A[2] = 4$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	10	8	6	4	2

$i = 10$, тогда $A[1] = A[10] = 2$, $A[10] = A[1] = 2$,

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
	2	4	6	8	10	10	8	6	4	2

Верный ответ: 2).

4.8.2.2. Описание одномерных массивов

Массив описывается в разделе переменных посредством указания типа его элементов и максимального их количества. Тип элементов может быть любым. При задании максимального числа элементов массива (числа ячеек таблицы) обычно употребляют диапазон целых чисел. Общий вид описания массива следующий:

```
VAR <Имя массива>: ARRAY[размерность] OF
<тип элементов>;
```

К имени массива предъявляются те же требования, что и к имени переменной.

Размерность — это диапазон целых чисел, каждое из которых является порядковым номером (индексом) одной из ячеек массива. Обычно диапазон индексов задают, начиная от единицы: [1..100]. Это удобно, но не обязательно. Важно лишь, чтобы правая граница диапазона была больше или равна левой границе.

```
Const Max=100;
Var A:Array[1..50] of Real; {массив "A" состоит из 50 вещественных чисел}
    B:Array[1..Max] of integer; {массив "B" состоит из ста целых чисел}
    Mas:Array[1999..2000] of integer; {массив "Mas" состоит из двух целых чисел}
    Mas2:Array[-700..-1] of Real; {массив "Mas2" состоит из 700 вещественных чисел}
```

.....

При описании массивов, задавая числовые пределы изменения индексов, обычно указывают максимально возможное число элементов, которое может быть востребовано в данной программе; при этом допускается, что фактическое количество элементов может оказаться меньше, чем затребовано. Но если значение индекса элемента массива не входит в указанный при описании диапазон, это приведет к ошибке.

4.8.2.3. Использование значений массивов, ввод и вывод значений линейных массивов

Как уже отмечалось, для ссылки на отдельные элементы массива используется переменная с индексом: имя массива[индекс]. Чтобы получить доступ к конкретному элементу массива, в качестве индекса можно использовать не только целое число, соответствующее порядковому номеру этого элемента в массиве, но и выражение, значение которого равно упомянутому целому числу. Например, при обращении к элементам некоторого массива “A: Array[1..100] of Real” в качестве индекса можно использовать любое арифметическое выражение, значением которого будет целое число из диапазона 1..100:

```
A[56]; A[i+7]; A[i div j].
```

Вводить и выводить значения из массивов целесообразно поэлементно, используя в цикле операторы Readln, Write, Writeln и оператор присваивания.

Пример 18. Составить программу, позволяющую ввести с клавиатуры массив из 20 элементов, а затем вывести из значения на экран.

Решение. При вводе и выводе элементов предусмотрим соответствующие комментарии: “A[1]=”, “A[2]=”, “A[3]=” и т.д.

```
Program Vvod_Vivod;  
Var i:integer;  
A:Array[1..20] of Real;  
B:Char;
```

```
Begin
writeln('Введите числа в массив: ');
for i:=1 to 20 do
begin
write('A['i,']= ');
readln(A[i])
end;
for i:=1 to 20 do
writeln('A['i,']='A[i]:6:2);
readln
End.
```

Пример 19. Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 50 — баллы учащихся за выполнение заданий на олимпиаде по информатике. Для получения возможности продолжить борьбу во втором туре участнику требовалось набрать не менее 15 баллов. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит минимальный балл среди учащихся, прошедших во второй тур олимпиады. Известно, что хотя бы один учащийся вышел во второй тур олимпиады.

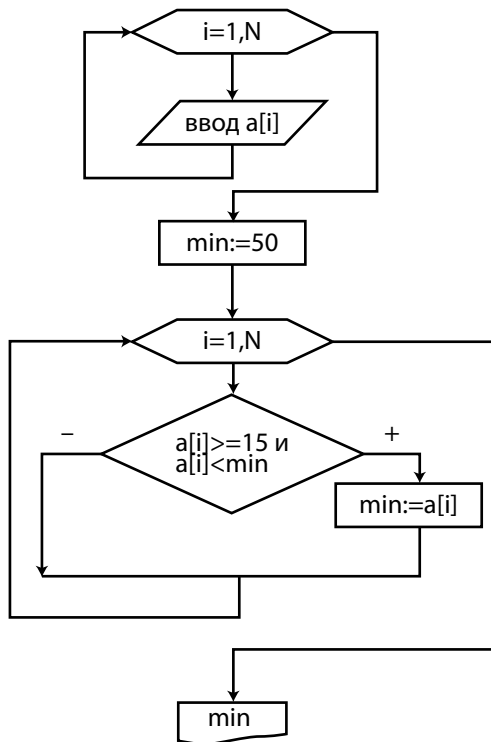
Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

```
const N=40;
var a: array [1..N] of integer;
i, j, min: integer;
begin
for i:=1 to N do readln(a[i]);
...
end.
```

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке програм-

мирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Решение. В силу того, что диапазон значений элементов массива известен (от 0 до 50), в качестве первоначального минимума выбираем наибольшее из возможных значений, то есть 50 ($\text{min}:=50$). Затем, просматривая все элементы массива, ищем те, которые, с одной стороны, по значению не меньше 15, а с другой стороны, меньше минимума. Найдя такое значение, меняем значение минимума. Блок-схема алгоритма выглядит следующим образом:



Текст программы имеет вид

```
const N=40;
var a: array [1..N] of integer;
i, j, min: integer;
begin
for i:=1 to N do readln(a[i]);
min:=50;
for i:=1 to N do
if (a[i]>=15) and (a[i]< min) then min:=a[i];
writeln('минимальный бал =',min)
end.
```

Пример 20. Опишите на русском языке или одном из языков программирования алгоритм подсчета максимального количества подряд идущих совпадающих элементов в целочисленном массиве длины 30.

Решение. Заведем переменную `MaxCount` для хранения максимального количества подряд идущих совпадающих элементов и счетчик `TCount` для хранения числа элементов в последней группе совпадающих элементов. Самая минимальная последовательность одинаковых элементов состоит из одного элемента, поэтому изначально полагаем `MaxCount = 1` и `TCount = 1`. Просматривая элементы массива, сравниваем очередной элемент со следующим за ним. Если значения совпадают, увеличиваем счетчик `TCount` на 1. Если очередной элемент массива оказывается не равным предыдущему, то сравниваем текущее значение счетчика со значением переменной `MaxCount`; если он больше, то заменяем значение переменной `MaxCount` значением счетчика. После сравнения записываем в счетчик `TCount` 1. Так повторяем до конца массива. В конце работы нужно еще раз сравнить значение счетчика `TCount` в последней последовательности со значением переменной `MaxCoin` и переопределить ее, если счетчик больше.

```
const N = 30;
var a:array[1..N] of integer;
MaxCount, TCount, i: integer;
begin
```



```
for i:=1 to N do readln(a[i]);
MaxCount: = 1;
TCount: = 1;
for i:= 2 to N do
begin
if a[i]=a[i-1] then TCount:=TCount+1;
else begin
if TCount > MaxCount then
MaxCount:=TCount;
TCount:=1;
end;
end;
if TCount > MaxCount then MaxCount:=TCount;
writeln(MaxCount);
end.
```

4.8.3. Двумерные массивы

Одномерный массив (вектор) имеет вид таблицы из одной строки, ячейки которой заполнены значениями. Рассматривая ДВУМЕРНЫЙ массив, можно провести аналогию с таблицей, имеющей несколько строк и столбцов. Если отдельный элемент одномерного массива мы обозначали именем массива с индексом ($A[i]$), то для обозначения элемента двумерного массива (матрицы) потребуются два индекса: один — для указания номера строки, другой — для указания номера столбца, на пересечении которых находится нужный элемент: $A[i,j]$. Двумерные массивы описываются сходно с массивами одномерными:

```
Const Str=4; Stolb=5;
Var A:Array[1..Str,1..Stolb] of Real;
```

здесь Str — число строк, $Stlb$ — число столбцов.

Заполнение двумерных массивов данными и вывод данных из двумерного массива осуществляются с использованием двух циклов, один из которых вложен в другой. Внутренний цикл перебирает индексы столбцов, наружный — индексы строк:

```

FOR i:=1 TO Str DO
FOR j:=1 TO Stlb DO Readln(A[i,j]);
.....
FOR i:=1 TO Str DO
FOR j:=1 TO Stlb DO Writeln('A['i',' ',j,']=',A
[i]:6:2);

```

Пример 21. Дан двумерный массив A размером $N \times N$. Требуется найти сумму элементов массива, расположенных на главной и побочной диагоналях.

$$\begin{pmatrix}
 a_{11} & a_{12} & a_{13} & a_{14} \\
 a_{21} & a_{22} & a_{23} & a_{24} \\
 a_{31} & a_{32} & a_{33} & a_{34} \\
 a_{41} & a_{42} & a_{43} & a_{44}
 \end{pmatrix}$$

Решение. В массиве более темным цветом отмечены элементы главной диагонали, они имеют индекс $A[i,i]$, $i = 1, 2, 3, \dots, N$. Более светлым тоном выделены элементы побочной диагонали, их индексы $A[i, N-i+1]$, $i = 1, 2, 3, \dots, N$. Сумма элементов, расположенных на диагоналях, вычисляется с помощью оператора цикла:

```

for i:=1 to N do
S:=S+A[i,i]+A[i,N-i+1];

```

Заметим, что если число N нечетное, то элемент, стоящий в середине массива, принадлежит и главной и побочной диагоналям, поэтому он просуммируется дважды во время выполнения цикла. В этом случае после завершения цикла необходимо вычесть средний элемент из суммы:

```

S:=S-A[N div 2+1, N div 2+1];

```

Полностью программа имеет вид:

```

Program N21;
Const N=5;
Var A :array[1..N,1..N] of real;
S :real;
i,j : integer;

```

```

begin
writeln('введите элементы массива');
for i:=1 to N do
for j:=1 to N do
begin
write('A[', i ,',', j ,']=');
readln(A[i,j]);
end;
for i:=1 to N do
S:=S+A[i,i]+A[i,N-i+1];
if N mod 2 <>0 then
S:=S-A[N div 2+1, N div 2+1];
writeln('S=',S:5:1);
readln;
End.

```

Пример 22. Дан целочисленный прямоугольный массив A размерности 5×8 . Опишите на русском или на одном из языков программирования алгоритм, который меняет местами столбцы, содержащие максимальный и минимальный элементы (предполагается, что такие элементы единственные).

Решение. Для решения этой задачи необходимо проделать следующие действия.

1) Определить минимальный элемент и номер столбца, в котором он находится. Для этого первоначально за минимум принимаем элемент $A[1,1]$ матрицы ($\text{min}:=A[1,1]$), соответственно номер столбца с минимальным элементом считаем равным 1 ($n_{\text{min}}:=1$). Затем, просматривая каждый элемент массива, сравниваем его с минимумом, и если он окажется меньше минимума, то меняем значение минимума и номера столбца n_{min} :

```

min:=A[i,j];
n_min:=j;

```

i меняется по строкам от 1 до 5, j меняется по столбцам от 1 до 8.

2) Определить максимальный элемент и номер столбца, в котором он находится. Кстати, искать максимум и минимум можно одновременно.

3) Поменять местами элементы столбцов с номерами n_min и n_max , используя для этого дополнительную ячейку.

```
t:=A[i,n_min];
A[i,n_min]:=A[i,n_max];
A[i,n_max]:=t;
i меняется по строкам от 1 до 5.
```

В результате получаем программу.

```
Program Z22;
var a:array[1..5,1..8] of integer;
min,n_min,max,n_max,i,j,t:integer;
begin
writeln('введите элементы массива');
for i:=1 to 5 do
for j:=1 to 8 do
begin
write('A[', i ,',', j ,']=');
readln(A[i,j]);
end;
writeln('был введен массив:');
for i:=1 to 5 do
begin
for j:=1 to 8 do
write(A[i,j]:4);
writeln;
end;
min:=A[1,1];n_min:=1;
max:=A[1,1];n_max:=1;
for i:=1 to 5 do
for j:=1 to 8 do
if A[i,j]<min then
begin
min:=A[i,j];
n_min:=j
end
else
if A[i,j]>max then
```

```
begin
max:=A[i,j];
n_max:=j
end;
for i:=1 to 5 do
begin
t:=A[i,n_min];
A[i,n_min]:=A[i,n_max];
A[i,n_max]:=t
end;
writeln('результат:');
for i:=1 to 5 do
begin
for j:=1 to 8 do
write(A[i,j]:4);
writeln;
end;
readln;
end.
```

4.8.4. Полная переменная

Массив можно рассматривать не только как группу однотипных переменных, но и как единое целое, как одну переменную. Значением этой особой переменной является весь массив. Такую переменную называют **ПОЛНОЙ**.

Если полные переменные **A** и **B** имеют один и тот же тип (были описаны одинаково) и всем элементам массива «**A**» были присвоены значения, то для присвоения полной переменной «**B**» точно такого же значения достаточно выполнить оператор присваивания **B:=A**.

4.9. СОЗДАНИЕ ПРОГРАММ ДЛЯ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ

Пример 23. Заключительный этап олимпиады по астрономии проводился для учеников 9–11-х классов, участвующих в общем конкурсе. Каждый участник олимпиады мог набрать от 0 до 50 баллов. Для определения побе-

дителей и призеров сначала отбираются 45% участников, показавших лучшие результаты.

По положению, в случае, когда у последнего участника, входящего в 45%, оказывается количество баллов такое же, как и у следующих за ним в итоговой таблице, решение по данному участнику и всем участникам, имеющим с ним равное количество баллов, определяется следующим образом: все участники признаются призерами, если набранные ими баллы больше половины максимально возможных; все участники не признаются призерами, если набранные ими баллы не превышают половины максимально возможных.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая по результатам олимпиады будет определять, какой минимальный балл нужно было набрать, чтобы стать победителем или призером олимпиады.

На вход программе сначала подается число участников олимпиады N . В каждой из следующих N строк находится результат одного из участников олимпиады в следующем формате:

<Фамилия> <Имя> <класс> <баллы> ,

где

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

<Имя> — строка, состоящая не более чем из 15 символов,

<класс> — число от 9 до 11,

<баллы> — целое число от 0 до 50 набранных участником баллов.

<Фамилия> и <Имя>, <Имя> и <класс>, а также <класс> и <баллы> разделены одним пробелом. Пример входной строки:

Иванов Петр 10 17

Программа должна выводить минимальный балл призера. Гарантируется, что хотя бы одного призера по указанным правилам определить можно.

Решение. Разобьем решение задачи на блоки и проанализируем действия каждого блока.

1. *Ввод и хранение исходных данных.* Исходными данными в этой задаче являются:

N — целое число, характеризующее число участников олимпиады,

N строк в формате: <Фамилия> <Имя> <класс> <баллы>.

Поскольку при постановке задачи сделан акцент на эффективности программы по используемой памяти, то проработки требуют следующие вопросы:

1. Нужно ли хранить все данные?
2. Как считать данные, если у них разный тип?
3. Где хранить исходные данные?

Так как требуется определить, какой минимальный балл нужно было набрать, чтобы стать победителем или призером олимпиады, то фамилии, имена и класс победителей и призеров для решения задачи не важны. Решающую роль имеют только баллы, набранные участниками, причем для нас важно, сколько участников олимпиады набрали то или иное количество баллов. Создадим массив `mb` (`var mb: array[0..50] of integer`), индексами в котором будут баллы от 0 до 50, а значениями — количество учеников, набравших данное количество баллов. Изначально значения элементов равны 0.

```
for i:=0 to 50 do
  mb[i]:=0;
```

Далее необходимо просмотреть все N строк исходной информации, выделить в них баллы участников и учесть их в массиве `mb`.

Сначала считаем фамилию посимвольно, до пробела:

```
repeat
  read(c);
until c=' '; {считана фамилия}
```

Затем посимвольно до пробела считываем имя:

```
repeat
  read(c);
until c=' '; {считано имя}
```

Теперь можем считать 2 числа: класс и баллы

```
readln(k,b);
```

и учесть, что b баллов набрал еще один участник олимпиады:

```
mb[b]:=mb[b]+1;
```

Полностью ввод информации выглядит следующим образом:

```
for i:=0 to 50 do
mb[i]:=0;
readln(N);
for i:=1 to N do
begin
repeat
read(c);
until c=' '; {считана фамилия}
repeat
read(c);
until c=' '; {считано имя}
readln(k,b);
mb[b]:=mb[b]+1;
end;
```

Необходимая информация введена и упорядочена по возрастанию баллов.

2. Обработка исходных данных (собственно решение задачи). Теперь требуется отобрать 45% участников, показавших лучшие результаты, и определить, какой минимальный балл нужно было набрать, чтобы стать победителем или призером олимпиады. Будем просматривать массив mb с конца и отсчитывать 45% участников: в переменную S будем складывать сначала число участников, набравших 50 баллов, затем 49 и т.д. до тех пор, пока не просуммируем 45% участников:

```
S:=0;
b:=50;
while S<=N*0.45 do
begin
```



```

S:=S+mb[b];
b:=b-1;
end;

```

Цикл завершится, когда $S > N * 0.45$, предыдущее значение b (т. е. $b+1$) и есть требуемый минимальный балл, однако возможна ситуация, когда никто из участников не набрал $b + 1$ баллов. Значит, вычисление S должно закончиться на шаг раньше:

```

S:=0;
b:=50;
while S+mb[b]<=N*0.45 do
begin
S:=S+mb[b];
if mb[b]>0 then minb:=b;
b:=b-1;
end;

```

Таким образом, определены те, кто наверняка стал призером, и пропущены баллы, которые никто не набрал. Теперь осталось учесть последнее условие: в случае, когда у последнего участника, входящего в 45% оказывается количество баллов такое же, как и у следующих за ним в итоговой таблице ($S < N * 0.45$), решение по данному участнику и всем участникам, имеющим с ним равное количество баллов, определяется следующим образом: все участники признаются призерами, если набранные ими баллы больше половины максимально возможных (25 баллов); все участники не признаются призерами, если набранные ими баллы не превышают 25.

```

if (S+1<=N*0.45) and (b>25) then minb:=b;

```

Все возможные ситуации проанализированы, и можно составить окончательный текст программы.

```

Program C23;
Var
mb: array[0..50] of integer;
c: char;
i, k, N, b, S, minb: integer;

```

```
Begin
for i:=0 to 50 do
mb[i]:=0;
readln(N);
for i:=1 to N do
begin
repeat
read(c);
until c=' '; {считана фамилия}
repeat
read(c);
until c=' '; {считано имя}
readln(k,b);
mb[b]:=mb[b]+1;
end;
S:=0;
b:=50;
while S+mb[b]<=N*0.45 do
begin
S:=S+mb[b];
if mb[b]>0 then minb:=b;
b:=b-1;
end;
if (S+1<=N*0.45) and (b>25) then minb:=b;
writeln(minb);
end.
```

Приведем критерии оценивания задания С4. Наиболее важные моменты при оценивании подчеркнуты.

4 балла: программа работает верно для всех входных данных. Входные данные размещены эффективно с точки зрения используемой памяти. Программа просматривает исходные данные только один раз. Искомые величины находятся путем однократного просмотра сохраненных данных. Допускается наличие в программе одной синтаксической ошибки.

3 балла: программа работает верно, однако отсутствует эффективность в хранении исходных данных и поис-

ке искомым величин (многократные просмотры исходных данных). Допускается наличие от одной до трех синтаксических ошибок.

2 балла: в целом программа работает верно, но в реализации алгоритма содержатся 1–2 ошибки. Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок.

1 балл: программа работает не при всех исходных данных. Допускается до 4 различных ошибок в реализации алгоритма. Допускается наличие от одной до семи синтаксических ошибок.

Пример 24. На автозаправочных станциях (АЗС) продается бензин с маркировкой 92, 95 и 98. В городе N был проведен мониторинг цены бензина на различных АЗС.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет определять для бензина с маркировкой 92, на какой АЗС его продают по второй по минимальности цене (считается, что самой низкой цене потребители не доверяют), а если таких АЗС несколько, то выдается только количество таких АЗС. Если все АЗС, у которых 92-й бензин есть, продают его по одной и той же цене, то эта цена считается искомой и выдается либо число таких АЗС, когда их несколько, либо конкретная АЗС, если она одна.

На вход программе в первой строке подается число данных о стоимости бензина N . В каждой из последующих N строк находится информация в следующем формате:

<Компания> <Улица> <Марка> <Цена>,

где

<Компания> — строка, состоящая не более чем из 20 символов без пробелов,

<Улица> — строка, состоящая не более чем из 20 символов без пробелов,

<Марка> — одно из чисел — 92, 95 или 98,

<Цена> — целое число в диапазоне от 1000 до 3000, обозначающее стоимость одного литра бензина в копейках.

<Компания> и <Улица>, <Улица> и <Марка>, а также <Марка> и <цена> разделены ровно одним пробелом.

Пример входной строки:

Лукойл Мичуринский 92 1950

Программа должна выводить через пробел <Компанию> и <Улицу> искомой АЗС или их количество, если искомых вариантов несколько.

Пример выходных данных: ТНК Можайский

Второй вариант выходных данных: 4

Решение. Рассмотрим следующие этапы решения задачи.

1. Ввод и хранение исходных данных. Исходными данными в этой задаче являются:

N — целое число, характеризующее число вводимых данных о стоимости бензина,

N строк в формате: <Компания> <Улица> <Марка> <Цена>.

Поскольку в задаче анализируются цены бензина с маркировкой 92, то всю остальную информацию можно не хранить в памяти. Так как ответ в ряде случаев требует информацию о <Компании> и <Улице>, то необходимо запоминать значение переменной (одной строкой) <Компания> + <Улица>, однако не для всех сведений о 92-м бензине, а лишь о сведениях с минимальной и по второй по минимальности цене.

2. Обработка исходных данных (собственно решение задачи). При обработке исходных данных требуется ответить на следующие вопросы:

1. Как определить вторую по минимальности цену?
2. Как определить количество таких АЗС?
3. Как определить, что на всех АЗС бензин продается по одной цене?

Рассмотрим вопрос нахождения второй по минимальности цены.

Первоначально в качестве минимума возьмем самую большую цену $\text{min1}:=3001$; информация о компании с такой ценой $\text{s1}:=$ ”; далее, просматривая подряд все сведения и отбирая только те, у которых марка 92, обозначим через b стоимость бензина, s – информацию о компании. Возможны следующие варианты:

если $b < \text{min1}$, то $\text{min2}:=\text{min1}$; $\text{s2}:=\text{s1}$; $\text{min1}:=b$; $\text{s1}:=s$;
здесь min2 — вторая по минимальности цена, s2 — информация о компании;

если $\text{min1} < b < \text{min2}$, то $\text{min2}:=b$; $\text{s2}:=s$;

Перейдем к вопросу о количестве АЗС со второй по минимальности ценой. Для того чтобы учитывать количество АЗС, заведем 2 переменные: cnt1 — количество АЗС, на которых минимальная цена на бензин, cnt2 — количество АЗС, на которых вторая по минимальности цена. Первоначально эти переменные равны 0. При обработке информации возможны следующие ситуации:

если $b < \text{min1}$, то $\text{cnt2}:=\text{cnt1}$; $\text{cnt1}:=1$;

если $b = \text{min1}$, то $\text{cnt1}:= \text{cnt1}+1$;

если $\text{min1} < b < \text{min2}$, то $\text{cnt2}:=1$;

если $b = \text{min2}$, то $\text{cnt2}:=\text{cnt2}+1$;

Понять, что на всех АЗС бензин продается по одной цене, можно, посмотрев значение переменной cnt2 : если $\text{cnt2}=0$, то на всех АЗС 92-й бензин продается по одной цене.

В итоге получаем программу:

```

Program Z24;
var c: char;
    i, k, N, b, min1, min2, cnt1, cnt2 :
integer;
    s, s1, s2: string;
begin
min1:=3001;
cnt1:=0;
readln(N);
for i:=1 to N do
begin
read(c);

```

```
s:='';
repeat
s:=s+c;
read(c);
until c=' '; {считана компания}
repeat
s:=s+c;
read(c);
until c=' '; {добавлена улица к компании}
readln(k,b);
if k=92 then {определение минимальных цен}
if min1 > b then
begin
min2:=min1; cnt2:=cnt1; s2:=s1;
min1:=b; cnt1:=1; s1:=s;
end else if min1 = b then cnt1:=cnt1+1
else if min2 > b then
begin
min2:=b; cnt2:=1; s2:=s;
end
else if min2 = b then cnt2:=cnt2+1;
end;
{вывод результата}
if cnt2>0 then
if cnt2=1 then writeln(s2)
else writeln(cnt2)
else if cnt1=1 then writeln(s1) else
writeln(cnt1)
end.
```

Пример 25. На вход программе подаются строчные английские буквы. Ввод этих символов заканчивается точкой (другие символы, отличные от «.» и букв «а»...«z», во входных данных отсутствуют). Требуется написать как можно более эффективную программу, которая будет печатать буквы, встречающиеся во входной последовательности, в порядке увеличения частоты их встречаемости. Каждая буква должна быть распечатана только 1 раз. Точка при этом не учитывается.

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

```
baobaba.
```

В данном случае программа должна вывести
oab

Решение. Рассмотрим два этапа решения задачи.

1. Ввод и хранение исходных данных. Исходными данными в этой задаче являются символы. Нам необходимо узнать частоту вхождения каждого символа. Для достижения этой цели воспользуемся двумя одномерными целочисленными массивами с индексами от 0 до 25 (по числу букв в латинском алфавите), в одном массиве значениями будут буквы от «а» до «z», в другом — частота вхождения этих букв.

```
Var
m: array[0..25] of 'a'..'z';
a: array[0..25] of integer;
```

Заполним значениями массив *m*:

```
for i:=0 to 25 do
m[i]:=chr(ord('a')+i)
```

Здесь функция `chr(·)` позволяет получить символьное значение (символ) по коду из таблицы ASCII, а функция `ord(·)` выполняет обратную операцию определения кода символа. В итоге элементы массива *m* примут значения `m[0]='a', m[1]='b', m[2]='c', m[3]='d', ... , m[25]='z'`.

Значения элементов массива *a* делаем равными 0:

```
for i:=0 to 25 do
a[i]:=0;
```

Оба присваивания можно осуществить в одном теле цикла

```
for i:=0 to 25 do
begin
a[i]:=0;
m[i]:=chr(ord('a')+i)
end;
```

Далее, считывая посимвольно исходную строку, подсчитываем частоту вхождения каждой буквы. Так, если считан символ «с», то он имеет порядковый номер $i:=\text{ord}(c)-\text{ord}('a')$, а значит, к значению элемента $a[i]$ нужно добавить 1, или $a[\text{ord}(c)-\text{ord}('a')]:=a[\text{ord}(c)-\text{ord}('a')+1]$;

2. Обработка исходных данных. Нам осталось лишь упорядочить массив a по возрастанию элементов, не забывая при перестановке элементов массива a одновременно переставлять соответствующие элементы массива m , чтобы не нарушить связь между элементами обоих массивов.

Для упорядочивания элементов массива воспользуемся методом «пузырька». Он выполняется в несколько шагов:

1-й шаг: каждый элемент с 0-го до предпоследнего, то есть до 24-го, сравнивается со своим соседом справа, и если он больше, чем правый элемент, то элементы меняются местами

```
if a[j] > a[j+1] then
begin
k:=a[j]; a[j]:=a[j+1]; a[j+1]:=k;
c:=m[j]; m[j]:=m[j+1]; m[j+1]:=c;
end;
```

В результате самый большой по значению элемент окажется на 25-м месте.

2-й шаг: проделываем то же самое с элементами с индексами от 0 до 23. В результате второй по величине элемент окажется на 24-м месте.

...

25-й шаг: в обмене участвуют только 0 и 1-й элементы, все остальные уже упорядочены.

В целом алгоритм сортировки имеет вид

```
for i:=24 downto 0 do
for j:=0 to i 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;
```


Далее выводим результат. Сначала пропускаем элементы, частота встреч которых равна 0, затем печатаем оставшиеся элементы массива *m*.

```
i:=0;
while a[i] = 0 do i:=i+1;
for j:=i to 25 do write(m[j]);
```

В итоге получаем программу:

```
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;
readln;
for i:=24 downto 0 do
for j:=0 to i do
if a[j] > a[j+1] then
begin
k:=a[j]; c:=m[j];
a[j]:=a[j+1]; m[j]:=m[j+1];
a[j+1]:=k; m[j+1]:=c;
end;
i:=0;
while a[i] = 0 do i:=i+1;
for j:=i to 25 do write(m[j]);
readln
end.
```

4.10. ОТЛАДКА ПРОГРАММЫ В СРЕДЕ ТУРБО-ПАСКАЛЬ

4.10.1. Основные правила написания текста программы

Возможны два варианта работы по созданию программы.

Первый сеанс работы по написанию новой программы.

В этом случае надо войти в главное меню (F10), выбрать пункт «File», нажать «Enter», далее в подменю выделить слово «New», нажать «Enter». В этом случае составляемой программе будет автоматически присвоено имя «NONAME.PAS», которое будет высвечено в правом верхнем углу окна EDIT. Чтобы дать программе какое-либо иное имя, следует в пункте «File» главного меню выбрать подпункт «Write to» и в появившемся окне набрать желаемое имя программы с указанием имени диска и, если необходимо, имени каталога. Выполнив перечисленное, следует нажать «Enter», после чего окно EDIT очистится, в верхнем правом углу окна будет выведено заданное название программы, а в левом верхнем углу будет мерцать курсор, приглашая к вводу с клавиатуры текста новой программы.

Редактируется (модифицируется) старая программа.

В этом случае надо в пункте «File» главного меню выбрать подпункт «Load» (можно просто нажать клавишу F3), после чего появится небольшое окно, в котором надо указать имя файла, содержащего текст редактируемой программы (при необходимости, кроме имени файла, указывается имя диска и имя каталога). Если имя файла точно не известно, нужно задать имя диска и шаблон выбора группы файлов (например, C:*.PAS или A:*.*). На экране появится список хранящихся на диске файлов и светлый прямоугольник-указатель, который следует поместить клавишами управления курсора на нужное имя в списке файлов и нажать «Enter». В результате в окне EDIT появится текст вашей программы, и мигающий курсор

сор будет сигнализировать о готовности текстового редактора к модификации программы.

После окончания сеанса набора текста программы или его редактирования надо сохранить текст на магнитном диске, нажав F2.

4.10.2. Стиль программирования, облегчающий отладку

По мнению экспертов, около 90% рабочего времени программиста затрачивается на отладку программы и только 10% — на ее разработку и написание, поэтому лучшим средством облегчить неизбежную отладку является профилактика ошибок еще на этапе разработки программы и написания ее текста. Правильно написанная программа содержит меньше ошибок и значительно облегчает их поиск.

Существует ряд простых правил, которых желательно придерживаться при составлении программы.

Писать текст программы и отлаживать ее следует небольшими частями. Перед тем как использовать результаты работы одной части программы в другой ее части, следует добиться правильной работы первого блока.

Следует стремиться к простоте алгоритма программы. Главное, чтобы программа начала работать правильно, а уже потом можно украсить ее эффектными средствами ввода-вывода, добиться скорости работы и более рационального использования памяти.

Текст программы должен быть «рыхлым», не следует писать операторы компактно, вплотную друг к другу. Желательно, чтобы на строке было не более одного оператора. Поскольку отладчик Турбо-Паскаля работает построчно, такой подход облегчит локализацию ошибки. На одной строке в виде списка можно писать лишь те операторы, которые надежны в отношении отладки и в проверке не нуждаются: A:=10; B:=3.14; Str:='Строка символов'; C:=True;.

Выполняемые в программе действия надо как можно чаще сопровождать комментариями { }, (* *).

4.10.3. Основные виды ошибок программирования

В ходе отладки программы приходится сталкиваться с тремя видами ошибок.

Синтаксические ошибки — текст программы не отвечает требованиям языка Паскаль. Такие ошибки выявляются сразу при попытке компиляции и запуска программы.

Ошибки в алгоритме (семантические ошибки) — программа работает без сбоев, но результаты получаются неверные.

Ошибки 1-го вида помогает отыскать сам компьютер, приостанавливая работу и выводя на первой строке сообщение о характере ошибки. Более того, при компиляции неправильной программы Турбо-Паскаль автоматически загружает в окно редактора сбойный участок текста программы и помещает курсор вблизи обнаруженной ошибки.

Перечислим ошибки, наиболее часто допускаемые начинающими программистами, и разъясним сообщения Турбо-Паскаля, выдаваемые им при выявлении этих ошибок в процессе компиляции текста программы:

Identifier expected — в этом месте программы должен находиться идентификатор (имя программного объекта). Возможно, программист пытался использовать в качестве идентификатора слово, зарезервированное Паскалем для других целей.

Unknown identifier — идентификатор (имя переменной, константы, процедуры и т.п.) не был упомянут в разделе описаний программы.

Duplicate identifier — попытка дважды описать один и тот же идентификатор.

Syntax error — синтаксическая ошибка, например строка символов не была заключена в кавычки или неверно записано служебное слово.

Line too long — компилятор не может обрабатывать текст программы со строками длиннее 126 символов. Скорее всего, программист забыл поставить апостроф, закрывающий текстовую строку, записываемую в переменную или выводимую на экран.

Type identifier expected — не указан тип идентификатора.

Variable identifier expected — на этом месте в программе должна стоять переменная.

Error in type — объявление типа данных не может начинаться с этого символа.

Type mismatch — а) тип переменной, стоящей слева от знака присваивания, отличается от значения выражения, стоящего справа; б) при обращении к процедуре типы формального и фактического параметров не совпадают; в) переменная данного типа не может служить индексом массива.

Begin expected — нужен begin.

End expected — нужен end.

Inte expression expected — требуется выражение типа Integer.

Boolean expression expected — требуется выражение типа boolean.

Do expected — пропущено слово “do”.

Of expected — пропущено слово “of”.

Then expected — пропущено слово “then”.

To expected — пропущено слово “to”.

String variable expected — требуется строковая переменная.

Error in expression — данный символ не может участвовать в выражении таким образом.

Division by zero — деление на ноль.

Constant and case types do not math — тип меток и тип селектора в операторе CASE не соответствуют друг другу.

Label not within current block — оператор GOTO не может ссылаться на метку, находящуюся вне текущего модуля.

Label already defined — такая метка уже есть.

Floating point overflow operation — величина вещественного числа вышла за пределы диапазона, допускаемого Паскалем (обычно при делении на ноль).

Invalid procedure or function reference — неправильный вызов подпрограммы.

4.10.4. Средства, которыми располагает отладчик Турбо-Паскаля

Отладчик используется в том случае, если программа допускает сбой при работе или дает неправильный результат. Отладчик позволяет выполнять программу построчно, отдельными блоками, дает возможность следить за изменениями значений переменных в ходе работы программы.

Трассирование (клавиши F7 или F8).

С помощью трассирования можно выполнить одну строку программы, остановиться и просмотреть результаты расчета. Если в программе есть процедура, то ее можно выполнить сразу за один шаг (F8) или трассировать построчно (F7).

Выполнение до курсора (F4).

Можно перевести курсор на некоторую строку программы, а затем, нажав F4, указать отладчику, чтобы он выполнил программу от начала до данной строки. При трассировании этот режим позволяет пропустить выполнение циклов и других малоинтересных частей программы и сразу перейти к той точке, где требуется начать отладку.

Вычисление и модификация (CTRL-F4).

Для эпизодической проверки значений переменных на экран можно выводить окно вычислений EVALUATE. С помощью этого окна можно также изменять текущее значение той или иной переменной. Отменить все задействованные средства отладки (кроме точек останова) можно клавишами CTRL-F2.

Просмотр (CTRL-F7).

В окно просмотра WATCH можно поместить имена переменных, встречающихся в программе. По мере выполнения программы по шагам значения, выводимые в окне просмотра рядом с именами соответствующих им переменных, будут меняться.

Прерывание выполнения программы (CTRL-F8).

Отдельные строки программы можно пометить как точки останова. Программа, запущенная на выполнение, доходит до помеченной строки и останавливается. После этого можно проверить значения переменных, начать трассировку или запустить программу до следующей точки останова. Выполнение программы можно также прерывать в любое время нажатием клавиш CTRL-BREAK.

Точкой останова можно сделать любую строку программы. Для этого на нужную строку устанавливается курсор и нажимается CTRL-F8, после чего строка будет выделена красной подсветкой. Программа запускается нажатием CTRL-F9 и выполняется обычным способом. При достижении точки останова программа останавливается, на экране появляется окно редактора «EDIT» с участком текста, где находится точка останова. Если в окне просмотра WATCH были помещены имена переменных, они будут сопровождаться своими текущими значениями. После останова программы можно воспользоваться любыми средствами отладки: можно продолжить программу трассированием или выполнением до курсора (F7, F8, F4). Можно проверять и изменять значения переменных (CTRL-F4), добавлять и удалять выражения из окна просмотра, устанавливать или отменять точки останова, можно просмотреть выводимую программой на экран информацию (ALT-F5). Чтобы начать выполнение программы сначала, надо нажать CTRL-F2, CTRL-F9. Продолжить программу до следующей точки останова — CTRL-F9. Чтобы отменить точку останова, надо поместить на нее курсор и нажать CTRL-F8. Для просмотра всех точек останова в программе надо несколько раз воспользоваться опцией VIEW NEXT BREAKPOINT из пункта главного меню BREAK/WATCH. При этом точки останова будут циклически, одна за другой, появляться в окне EDIT вместе с близлежащими участками текста.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Программирование

1. Известны катеты прямоугольного треугольника a и b . Требуется составить программу, которая будет вычислять гипотенузу c и площадь S прямоугольного треугольника.

2. Известна высота цилиндра H и радиус его основания R . Требуется составить программу, которая будет вычислять объем V и площадь полной поверхности S цилиндра.

3. Определите значение переменной b после выполнения следующего фрагмента программы, где a и b — вещественные (действительные) переменные:

```
a := -5;
b := 5 + 7 * a;
b := b / 2 * a;
```

1) 3 2) -3 3) 75 4) -75

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

```
a:=5;
a:=a+6;
b:= -a;
c:=a-2*b;
```

1) $c = -11$
 2) $c = 15$
 3) $c = 27$
 4) $c = 33$

5. Даны стороны треугольника a , b , c . Требуется составить программу, которая будет вычислять площадь треугольника S .

6. Определите значение целочисленных переменных x и y после выполнения фрагмента программы:

```
x := 336;
y := 8;
x := x div y;
y := x mod y;
```

- 1) $x = 42, y = 2$ 3) $x = 2, y = 24$
 2) $x = 36, y = 12$ 4) $x = 24, y = 4$

7. Переменные a и b описаны в программе как целочисленные. Определите значение переменной a после выполнения следующего фрагмента программы $a:=117$;

```
b:=a mod 100;
a:=a div 2;
a:=a + b mod 2;
```

- 1) 9 2) 2 3) 59,5 4) 6,5

8. Даны 2 числа. Требуется составить программу, которая будет находить наибольшее из них.

9. Даны координаты 3 точек. Составить программу, которая будет определять, какая из них ближе к началу координат.

10. Составить программу, которая при любых значениях «а» и «b» решает уравнение $ax + b = 0$.

11. Требовалось написать программу, которая решает уравнение $ax^2 + bx + c = 0$ относительно x для действительных чисел a, b, c , введенных с клавиатуры, о которых заведомо известно, что $a \neq 0, b \neq 0$ и $c \neq 0$. Была написана следующая программа:

```
var a, b, c, D, x1, x2: real;
begin
  readln(a, b, c, x1, x2);
  D := b*b - 4*a*c;
  if D > 0 then
```

```

begin
x1 := (-b + sqrt(D))/(2*a);
x2 := (-b - sqrt(D))/(2*a);
write('x1 =', x1);
write('x2 =', x2);
end
else writeln ('действительных корней нет');
end.

```

Известно, что программа написана с ошибками. Последовательно выполните три задания:

- 1) Приведите пример таких чисел a , b , c , при которых программа неверно решает поставленную задачу.
- 2) Укажите, какая часть программы является лишней.
- 3) Укажите, как, по вашему мнению, нужно доработать программу, чтобы не было случаев ее неправильной работы.

12. Требовалось написать программу, которая определяет, лежит ли точка $A(x, y)$ внутри некоторого кольца («внутри» понимается в строгом смысле, т.е. случай, когда точка A лежит на границе кольца, недопустим). Центр кольца находится в начале координат. Для кольца заданы внутренний и внешний радиусы r_1 , r_2 ; известно, что r_1 отлично от r_2 , но неизвестно, $r_1 > r_2$ или $r_2 > r_1$. В том случае, когда точка A лежит внутри кольца, программа должна выводить соответствующее сообщение, в противном случае никакой выходной информации не выдается. Программист сделал в программе ошибки.

```

var x, y, r1,r2:real;
h: real;
begin
readln (x, y, r1, r2);
h := sqrt (x*x + y*y);
if (h < r1) and (h > r2) then
writeln ('точка A лежит внутри кольца');
end.

```

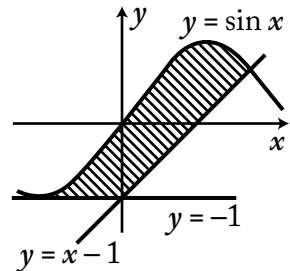
Последовательно выполните задания:

1) Приведите пример таких чисел x и 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.)

3) Укажите, как можно доработать программу, чтобы она не содержала логических операций **and** или **or**.

13. 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. Вот она:



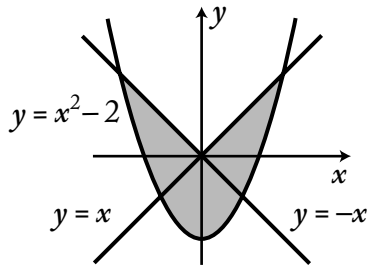
```
var x,y: real;
begin
  readln(x,y);
  if y >= -1 then
    if y <= sin(x) then
      if y >= x-1 then
        write('принадлежит')
      else
        write('не принадлежит')
    end.
```

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.)

14. 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. Вот она:



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

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.)

15. Дано действительное число x . Составить программу, которая будет вычислять сумму

$$S = \frac{x^2}{1} + \frac{x^2}{2} + \frac{x^2}{3} + \dots + \frac{x^2}{20}.$$

16. Написать программу, которая будет вычислять

$$Z = \sqrt{3 + \sqrt{6 + \dots + \sqrt{96 + \sqrt{99}}}}.$$

17. Дано натуральное число N и цифра k . Составить программу, которая будет определять, входит ли цифра k в запись числа N .

18. Дано натуральное число N . Составить программу, которая будет менять порядок цифр в записи числа N .

19. Дан целочисленный одномерный массив. Напишите программу, которая будет определять количество и произведение положительных элементов массива. Если таких элементов нет, то выдать сообщение об этом.

20. Дан целочисленный одномерный массив. Напишите программу, которая будет печатать те элементы массива, которые больше своих соседей. Если таких элементов нет, то выдать сообщение об этом.

21. Дан вещественный одномерный массив. Напишите программу, которая будет определять, составляют ли элементы массива возрастающую последовательность.

22. Значения элементов двух массивов A и B размером 1×100 задаются с помощью следующего фрагмента программы:

```
for i:=1 to 100 do
  A[i] := 50 - i;
for i:=1 to 100 do
  B[i] := A[i] + 49;
```

Сколько элементов массива B будут иметь отрицательные значения?

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

23. Значения элементов двумерного массива A были равны 0. Затем значения некоторых элементов были изменены (см. представленный фрагмент программы): $n := 0$;

```
for i:=1 to 5 do
  for j:=1 to 6-i do begin
    n := n + 1;
    A[i,j] := n;
  end;
```

Какой элемент массива будет иметь в результате максимальное значение?

- 1) A[1,1]
- 2) A[1,5]
- 3) A[5,1]
- 4) A[5,5]

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

```
for n:=1 to 5 do
for k:=1 to 5 do
V[n,k]:=n+k;
```

Чему будет равно значение V(2,4)?

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

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

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

Чему окажутся равны элементы этого массива?

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

26. Значения элементов двумерного массива A размером 5×5 задаются с помощью вложенного цикла в представленном фрагменте программы:

```
for i:=1 to 5 do
  for j:=1 to 5 do begin
    A[i,j] := i*j;
  end;
```

Сколько элементов массива будут иметь значения больше 10?

- 1) 8 2) 12 3) 10 4) 4

27. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

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

Как изменяются элементы этого массива?

- 1) Все элементы, кроме последнего, окажутся равны между собой.
2) Все элементы окажутся равны своим индексам.
3) Все элементы, кроме последнего, сдвигаются на один элемент вправо.
4) Все элементы, кроме последнего, уменьшаются на единицу.

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

прецедента использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

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

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия.

29. Дан целочисленный прямоугольный массив 6×10 . Опишите на русском языке или на одном из языков программирования алгоритм вычисления суммы минимальных элементов из каждой строки. Напечатать значение этой суммы. Предполагается, что в каждой строке минимальный элемент единственный.

30. Дан целочисленный массив из 31 элемента, в котором записаны значения температуры воздуха в марте. Элементы массива могут принимать значения от (-20) до 20. Опишите алгоритм, который подсчитывает и выводит среднюю температуру по всем дням, когда была оттепель (температура поднималась выше нуля). Гарантируется, что хотя бы один день в марте была оттепель. Исходные данные объявлены так, как показано ниже. Использовать другие переменные запрещается.

```
const N = 31;
var A: array[1..N] of integer;
i, x, y: integer;
s: real;
begin
for i:=1 to N do readln(A[i]);
...
end.
```


31. Дан целочисленный массив из 30 элементов. Элементы массива могут принимать произвольные целые значения. С клавиатуры вводится целое число X . Опишите алгоритм, который находит и выводит наименьший номер элемента, равного X , или сообщение, что такого элемента нет. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

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

32. На автозаправочных станциях (АЗС) продается бензин с маркировкой 92, 95 и 98. В городе N был проведен мониторинг цены бензина на различных АЗС.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет определять для каждого вида бензина, сколько АЗС продают его дешевле всего. На вход программе в первой строке подается число данных о стоимости бензина. В каждой из последующих N строк находится информация в следующем формате:

<Компания> <Улица> <Марка> <Цена>

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

Синойл Цветочная 95 2250

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

12 1 0

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

Иванов Петр 4 5 4

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

34. Вступительные испытания в некоторый вуз состоят из трех экзаменов: математика (максимальный балл — 9), информатика (максимальный балл — 9), литература (максимальный балл — 5). На вход программе подаются сведения о сдаче этих экзаменов абитуриентами. В первой строке вводится количество абитуриентов N , во второй — количество мест K ($K < N$), на которые эти абитуриенты претендуют. Каждая из следующих N строк имеет следующий формат: <Фамилия> <оценка1>

<оценка2> <оценка3>, где <Фамилия> — строка, состоящая не более чем из 20 символов, оценки — числа от 0 до максимальной оценки по предмету соответственно. (Ноль ставится в случае, если экзамен не сдавался, например, после полученной на предыдущем экзамене двойки. Все баллы, большие 2, считаются удовлетворительными.)
Пример входных строк:

```
Иванов 8 9 3
Петров 2 0 0
```

Требуется написать программу на языке Паскаль или Бейсик, которая определяла бы по имеющимся данным количество абитуриентов, набравших полупроходной балл в данный вуз, или сообщала, что такой балл отсутствует. (Полупроходным называется такой балл, что лишь часть абитуриентов, набравших такой балл и не получивших ни одной неудовлетворительной оценки, попадает в К лучших, которые должны быть зачислены на 1-й курс.) Считается, что абитуриенты, получившие только удовлетворительные оценки, обязательно присутствуют.

35. На вход программе подается набор символов, заканчивающийся точкой (в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введена точка, или считывать данные из файла). Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая сначала будет определять, есть ли в этом наборе символы, соответствующие десятичным цифрам. Если такие символы есть, то можно ли переставить их так, чтобы полученное число было симметричным (читалось одинаково как слева направо, так и справа налево). Ведущих нулей в числе быть не должно, исключение — число 0, запись которого содержит ровно один ноль. Если требуемое число составить невозможно, то программа должна вывести на экран слово «NO». А если возможно, то в первой строке следует вывести слово «YES», а во второй — искомое

симметричное число. Если таких чисел несколько, то программа должна выводить максимальное из них. Например, пусть на вход подаются следующие символы:

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

В данном случае программа должна вывести

```
YES  
91019
```

СПИСОК ЛИТЕРАТУРЫ

1. *Гусева И.Ю.* Информатика: Раздаточный материал тренировочных тестов. — СПб.: Тритон, 2008.
 2. *Зуев Е.А.* Язык программирования Turbo Pascal 6.0. — М.: Унитех, 1992.
 3. *Семакин И.Г.* и др. Информатика: Задачник-практикум в 2 т. / Под ред. И.Г. Семакина, Е.К. Хеннера. — М.: БИНОМ. Лаборатория знаний, 2007.
 4. *Солодовников А.Б.* Алгоритмические языки и основы программирования. Курс лекций в 2 частях.
 5. *Шауцукова Л.З.* Информатика: Учебное пособие для 10–11-х классов общеобразовательных учреждений. — М.: Просвещение, 2003.
 6. *Шаповалова И.А.* Заочная подготовка к ЕГЭ по информатике. — Тверь: Тверской государственный университет, 2011.
 7. *Ярцева О.Я., Цикина Е.Н.* Информатика: ЕГЭ-2010: Самые новые задания. — М.: АСТ: Астрель, 2010.
 8. Демонстрационные варианты ЕГЭ 2004–2010 гг.
 9. <http://kpolyakov.narod.ru>
 10. <http://olymp.ifmo.ru>
- [http://edu.dvgups.ru/MetDoc/Its/Izisk/ALang/Html/main.htm#А.Б. Солодовников.](http://edu.dvgups.ru/MetDoc/Its/Izisk/ALang/Html/main.htm#А.Б.Солодовников)

Издание для дополнительного образования

Для старшего школьного возраста

В ПОМОЩЬ СТАРШЕКЛАССНИКУ

**Самылкина Надежда Николаевна
Сильченко Ален Павлович**

ИНФОРМАТИКА
все темы для подготовки к ЕГЭ

Директор редакции *Л. Бершидский*
Ответственный редактор *А. Жилинская*
Ведущий редактор *Т. Судакова*
Художественный редактор *Н. Кудря*
Технический редактор *Л. Зотова*
Компьютерная верстка *Г. Ражикова*
Корректор *Е. Щукина*

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

Подписано в печать 05.08.2011.
Формат 60×90¹/₁₆. Гарнитура «Школьная».
Печать офсетная. Бумага тип. Усл. печ. л. 13,0.
Тираж экз. Заказ

ISBN 978-5-699-42984-4



9 785699 429844 >