

**Е. С. Павлова, И. Г. Лемешкина  
И. Л. Гоник, Е. В. Кузьмин, Ю. В. Левин**

# **ИНФОРМАТИКА**

**Волгоград  
2021**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Е. С. Павлова, И. Г. Лемешкина  
И. Л. Гоник, Е. В. Кузьмин, Ю. В. Левин

# ИНФОРМАТИКА

*Учебное пособие для поступающих в ВолгГТУ*



Волгоград  
2021

УДК 681.3 (075)

Р е ц е н з е н т :

канд. пед. наук доцент кафедры «Теория и методика обучения  
физике и информатике» ВГПУ *Л. Ю. Кравченко*

Печатается по решению редакционно-издательского совета  
Волгоградского государственного технического университета

**Павлова, Е. С.**

Информатика: учебное пособие для поступающих в ВолгГТУ/  
Е. С. Павлова, И. Г. Лемешкина, И. Л. Гоник, Е. В. Кузьмин,  
Ю. В. Левин; ВолгГТУ. – Волгоград, 2021. – 28 с.

ISBN 978-5-9948-4173-0

Пособие содержит учебно-тренировочные тесты, составленных с учетом реальных вариантов ЕГЭ прошлых лет, а также некоторые сведения из информатики, знание которых позволит избежать ошибок, обычно допускаемых учащимися при решении задач.

Предназначено для абитуриентов, учащихся выпускных классов школ, лицеев, гимназий, колледжей, слушателей факультета довузовской подготовки ВолгГТУ.

Библиогр.: 5 назв.

ISBN 978-5-9948-4173-0

© Волгоградский государственный  
технический университет, 2021

## ОГЛАВЛЕНИЕ

Введение .....	4
1. Системы счисления и двоичное представление информации в памяти компьютера .....	4
2. Связь между системами счисления .....	5
3. Кодирование текстовой информации. Кодировка ASCII. Основные кодировки кириллицы. Измерение количества информации .....	7
4: Вычисление информационного объема сообщения .....	8
5. Определение скорости передачи информации при заданной пропускной способности канала .....	100
6. Построение таблиц истинности логических выражений .....	111
7. Электронные таблицы .....	122
8. Составление запросов для поисковых систем с использованием логических выражений .....	15
9. Программирование базовых алгоритмических структур .....	17
10. Алгоритмы обработки одномерных массивов .....	233
Список используемой литературы .....	27

## Введение

Пособие содержит задания, составленные по структуре прошлых лет.

Пособие поможет выпускникам и абитуриентам самостоятельно подготовиться к успешной сдаче не только к сдаче внутренних вступительных испытаний в ВУЗ, но и к ЕГЭ.

Желаем успеха!

### 1. Системы счисления и двоичное представление информации в памяти компьютера

Системой счисления называется способ изображения чисел с помощью ограниченного набора символов, имеющих определенные количественные значения. Систему счисления образуют совокупность правил и приемов представления чисел с помощью набора знаков (цифр).

Различают позиционные и непозиционные системы счисления. В позиционных системах каждая цифра имеет определенный вес, зависящий от позиции цифры в последовательности, изображающей число. Позиция цифры называется разрядом. В позиционной системе счисления любое число можно представить в виде:

$$A_N = \sum_{i=-k}^{m-1} a_i * N^i$$

где

$a_i$  –  $i$ -ая цифра числа,

$k$  – количество цифр в дробной части числа,

$m$  – количество цифр в целой части числа,

$N$  – основание системы счисления.

Основание системы счисления показывает, во сколько раз «вес»  $i$ -го разряда больше ( $i-1$ ) разряда. Целая часть числа отделяется от дробной части точкой (запятой).

Например,  $11001,01_2 = 1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 + 0*2^{-1} + 1*2^{-2} = 25,75$

Представление чисел в различных системах счисления допускает однозначное преобразование их из одной системы счисления в другую.

Целое число с основанием  $N_1$  переводится систему счисления с основанием  $N_2$  путем последовательного деления числа  $A_{N_1}$  на основание  $N_2$ , записанного в виде числа с основанием  $N_1$ , до получения остатка. Полученное частное следует вновь делить на основание  $N_2$ , и этот процесс нужно повторять до тех пор, пока частное не станет меньше делителя. Полученные остатки от деления и последнее частное записываются в поряд-

ке, обратном полученному от деления. Сформированное число и будет являться числом с основанием N2.

Для примера переведем десятичное число  $A_{10}=37$  в двоичную систему счисления:

37 делим на 2 – частное 18, остаток 1,  
18 делим на 2 - частное 9, остаток 0,  
9 делим на 2 – частное 4, остаток 1,  
4 делим на 2- частное 2, остаток 0,  
2 делим на 2, частное 1, остаток 0.

Теперь записываем последнее частное и остатки от деления в обратном порядке. В результате получим  $A_2=100101$ .

### Пример задания

Сколько единиц в двоичной записи числа 1025?

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

**Решение (вариант 1, прямой перевод):**

1) переводим число 1025 в двоичную систему:  $1025 = 10000000001_2$

- 2) считаем единицы, их две  
3) Ответ: 2

**Решение (вариант 2, разложение на сумму степеней двойки):**

1) тут очень полезно знать наизусть таблицу степеней двойки, где  $1024 = 2^{10}$  и  $1 = 2^0$

2) таким образом,  $1025 = 1024 + 1 = 2^{10} + 2^0$

3) вспоминая, как переводится число из двоичной системы в десятичную (значение каждой цифры умножается на 2 в степени, равной её разряду), понимаем, что в двоичной записи числа ровно столько единиц, сколько в приведенной сумме различных степеней двойки, то есть, 2.

- 4) Ответ: 2

## 2. Связь между системами счисления

Так как двоичная, восьмеричная и шестнадцатеричная системы счисления связаны между собой через степени числа 2, то преобразования между ними можно выполнять другим более простым способом.

Для перевода из шестнадцатеричной (восьмеричной) системы счисления в двоичную достаточно двоичным кодом записать шестнадцатерич-

ные цифры тетрадами (по четыре двоичных разряда) и триадами (по три двоичных разряда) – для восьмеричных цифр.

Обратный перевод из двоичного кода производится в обратном порядке: двоичное число разбивается влево и вправо от границы целой и дробной части на тетрады – для последующей записи цифр в шестнадцатеричном представлении, на триады – для записи их значений восьмеричными цифрами.

### Пример задания

*Дано:  $a = D7_{16}$  и  $b = 331_8$ . Какое из чисел  $c$ , записанных в двоичной системе счисления, удовлетворяет неравенству  $a < c < b$ ?*

- 1)  $11011001_2$
- 2)  $11011100_2$
- 3)  $11010111_2$
- 4)  $11011000_2$

#### Общий подход:

перевести все числа (и исходные данные, и ответы) в одну (любую!) систему счисления и сравнить.

#### Решение (вариант 1, через десятичную систему):

- 1)  $a = D7_{16} = 13 \cdot 16 + 7 = 215$
- 2)  $b = 331_8 = 3 \cdot 8^2 + 3 \cdot 8 + 1 = 217$
- 3) переводим в десятичную систему все ответы:  
 $11011001_2 = 217$ ,  $11011100_2 = 220$ ,  $11010111_2 = 215$ ,  $11011000_2 = 216$
- 4) очевидно, что между числами 215 и 217 может быть только 216
- 5) таким образом, верный ответ – 4.

#### Решение (вариант 2, через двоичную систему):

- 1)  $a = D7_{16} = 1101\ 0111_2 = 11010111_2$  (каждая цифра шестнадцатеричной системы *отдельно* переводится в четыре двоичных – *тетраду*);
- 2)  $b = 331_8 = 011\ 011\ 001_2 = 11011001_2$  (каждая цифра восьмеричной системы *отдельно* переводится в три двоичных – *триаду*, старшие нули можно не писать);
- 3) между этими числами находится только двоичное число  $11011000_2$  – это ответ 4.

#### Решение (вариант 3, через восьмеричную систему):

- 1)  $a = D7_{16} = 11010111_2 = 011\ 010\ 111_2 = 327_8$  (сначала перевели в двоичную систему, потом двоичную запись числа разбили на триады **справа**

**налево**, каждую триаду перевели *отдельно* в десятичную систему, так как для чисел от 0 до 7 их восьмеричная запись совпадает с десятичной);

2)  $b = 331_8$ , никуда переводить не нужно;

3) переводим в восьмеричную систему все ответы:

$11011001_2 = 011 \ 011 \ 001_2 = 331_8$  (разбили на триады **справа налево**, каждую триаду перевели *отдельно* в десятичную систему, как в п. 1)

$11011100_2 = 334_8, 11010111_2 = 327_8, 11011000_2 = 330_8$

4) в восьмеричной системе между числами  $327_8$  и  $331_8$  может быть только  $330_8$

5) таким образом, верный ответ – 4 .

**Решение (вариант 4, через шестнадцатеричную систему):**

1)  $a = D7_{16}$  никуда переводить не нужно;

2)  $b = 331_8 = 11011001_2 = 1101 \ 1001_2 = D9_{16}$  (сначала перевели в двоичную систему, потом двоичную запись числа разбили на тетрады **справа налево**, каждую тетраду перевели в шестнадцатеричную систему; при этом тетрады можно переводить из двоичной системы в *десятичную*, а затем заменить все числа, большие 9, на буквы – A, B, C, D, E, F);

3) переводим в шестнадцатеричную систему все ответы:

$11011001_2 = 1101 \ 1001_2 = D9_{16}$  (разбили на тетрады **справа налево**, каждую тетраду перевели *отдельно* в десятичную систему, все числа, большие 9, заменили на буквы – A, B, C, D, E, F, как в п. 1)

$11011100_2 = DC_{16}, 11010111_2 = D7_{16}, 11011000_2 = D8_{16}$

4) в шестнадцатеричной системе между числами  $D7_{16}$  и  $D9_{16}$  может быть только  $D8_{16}$

5) таким образом, верный ответ – 4 .

### 3. Кодирование текстовой информации. Кодировка ASCII.

#### Основные кодировки кириллицы.

#### Измерение количества информации

При алфавитном подходе к кодированию текстовой информации все символы кодируются одинаковым числом бит. Чаще всего используют кодировки, в которых на символ отводится 8 бит (8-битные) или 16 бит (16-битные).

При измерении количества информации принимается, что в одном байте 8 бит, а в одном килобайте (1 Кбайт) – 1024 байта, в мегабайте (1 Мбайт) – 1024 Кбайта.

Чтобы найти информационный объем текста  $I$ , нужно умножить количество символов  $N$  на число бит на символ  $K$ :

$$I = N \cdot K$$

### Пример задания

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

#### Решение:

- 1) обозначим количество символов через  $N$
- 2) при 16-битной кодировке объем сообщения –  $16 \cdot N$  бит
- 3) когда его перекодировали в 8-битный код, его объем стал равен –  $8 \cdot N$  бит
- 4) таким образом, сообщение уменьшилось на  $16 \cdot N - 8 \cdot N = 8 \cdot N = 480$  бит
- 5) отсюда находим  $N = 480 / 8 = 60$  символов.

## 4. Вычисление информационного объема сообщения

С помощью  $K$  бит можно закодировать  $Q$  различных вариантов (чисел).

$$Q = 2^K$$

Таблица степеней двойки, показывает, сколько вариантов  $Q$  можно закодировать с помощью  $K$  бит:

K, бит	1	2	3	4	5	6	7	8	9	10
Q, вариантов	2	4	8	16	32	64	128	256	512	1024

При измерении количества информации принимается, что в одном байте 8 бит, а в одном килобайте (1 Кбайт) – 1024 байта, в мегабайте (1 Мбайт) – 1024 Кбайта.

### Пример задания

*В велокроссе участвуют 119 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного фи-*

ниша, 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?

**Решение:**

- 1) велосипедистов было 119, у них 119 разных номеров, то есть, нам нужно закодировать 119 вариантов
- 2) по таблице степеней двойки находим, что для этого нужно минимум 7 бит (при этом можно закодировать 128 вариантов, то есть, еще есть запас); итак, 7 бит на один отсчет
- 3) когда 70 велосипедистов прошли промежуточный финиш, в память устройства записано 70 отсчетов
- 4) поэтому в сообщении  $70 \cdot 7 = 490$  бит информации.

**Пример задания**

В некоторой стране автомобильный номер длиной 7 символов составляется из заглавных букв (всего используется 26 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 20 автомобильных номеров.

**Решение:**

- 1) всего используется 26 букв + 10 цифр = 36 символов;
- 2) для кодирования 36 вариантов необходимо использовать 6 бит, так как  $2^5 = 32 < 36 \leq 2^6 = 64$ , т.е. пяти бит не хватит (они позволяют кодировать только 32 варианта), а шести уже достаточно;
- 3) таким образом, на каждый символ нужно 6 бит (минимально возможное количество бит);
- 4) полный номер содержит 7 символов, каждый по 6 бит, поэтому на номер требуется  $6 \cdot 7 = 42$  бита;
- 5) по условию каждый номер кодируется целым числом байт (в каждом байте – 8 бит), поэтому требуется 6 байт на номер ( $5 \cdot 8 = 40 < 42 \leq 6 \cdot 8 = 48$ ), пяти байтов не хватает, а шесть – минимально возможное количество;
- 6) на 20 номеров нужно выделить  $20 \cdot 6 = 120$  байт.

## 5. Определение скорости передачи информации при заданной пропускной способности канала

Объем переданной информации  $Q$  вычисляется по формуле

$$Q = q \cdot t,$$

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

При вычислениях нужно помнить, что соотношение между единицами измерения количества информации представляют собой степени двойки:

$$1 \text{ байт} = 8 \text{ бит} = 2^3 \text{ бит},$$

$$1 \text{ Кбайт} = 1024 \text{ байта} = 2^{10} \text{ байта} = 2^{13} \text{ бит},$$

$$1 \text{ Мбайт} = 1024 \text{ Кбайта} = 2^{10} \text{ Кбайта} = 2^{10} \text{ байта} = 2^{23} \text{ бит}.$$

Правила выполнения операций со степенями:

при умножении степени при одинаковых основаниях складываются

$$2^a \cdot 2^b = 2^{a+b}$$

... а при делении – вычитаются:

$$\frac{2^a}{2^b} = 2^{a-b}$$

### Пример задания

Скорость передачи данных через ADSL-соединение равна 128000 бит/с. Через данное соединение передают файл размером 625 Кбайт. Определите время передачи файла в секундах.

#### Решение:

1) выделим в заданных больших числах степени двойки и переведем размер файла в биты, чтобы «согласовать» единицы измерения:

$$q = 128000 \text{ бит/с} = 128 \cdot 1000 \text{ бит/с} = 2^7 \cdot 125 \cdot 8 \text{ бит/с} = 2^7 \cdot 5^3 \cdot 2^3 \text{ бит/с} = 2^{10} \cdot 5^3 \text{ бит/с}$$

$$Q = 625 \text{ Кбайт} = 5^4 \text{ Кбайт} = 5^4 \cdot 2^{13} \text{ бит}$$

2) чтобы найти время передачи в секундах, нужно разделить размер файла на скорость передачи:

$$t = \frac{Q}{q} = \frac{5^4 \cdot 2^{13} \text{ бит}}{5^3 \cdot 2^{10} \text{ бит/с}} = 5 \cdot 2^3 \text{ с} = 40 \text{ с}$$

3) таким образом, ответ – 40 с.

## 6. Построение таблиц истинности логических выражений

Условные обозначения логических операций

$\neg A, \bar{A}$  не  $A$  (отрицание, инверсия)

$A \wedge B, A \cdot B$   $A$  и  $B$  (логическое умножение, конъюнкция)

$A \vee B, A + B$   $A$  или  $B$  (логическое сложение, дизъюнкция)

$A \rightarrow B$  импликация (следование)

Операцию «импликация» можно выразить через «ИЛИ» и «НЕ»:

$A \rightarrow B = \neg A \vee B$  или в других обозначениях  $A \rightarrow B = \bar{A} + B$

Если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ», и самая последняя – «импликация»

Таблица истинности выражения определяет его значения при всех возможных комбинациях исходных данных

Количество *разных* логических выражений, удовлетворяющих неполной таблице истинности, равно  $2^k$ , где  $k$  – число *отсутствующих* строк; например, полная таблица истинности выражения с тремя переменными содержит  $2^3=8$  строчек, если заданы только 6 из них, то можно найти  $2^{8-6}=2^2=4$  *разных* логических выражения, удовлетворяющие этим 6 строкам (но отличающиеся в двух оставшихся)

### Сводная таблица истинности основных логических операций

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

### Пример задания

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

Дан фрагмент таблицы истинности выражения  $F$ :

Какое выражение соответствует  $F$ ?

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

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

**Решение:**

- 1) для каждой строчки нужно подставить заданные значения  $X$ ,  $Y$  и  $Z$  во все функции, заданные в ответах, и сравнить результаты с соответствующими значениями  $F$  для этих данных;
- 2) если для какой-нибудь комбинации  $X$ ,  $Y$  и  $Z$  результат не совпадает с соответствующим значением  $F$ , оставшиеся строчки можно не рассматривать, поскольку для правильного ответа все три результата должны совпасть со значениями функции  $F$ ;
- 3) перепишем ответы в других обозначениях:
  - 1)  $\bar{X} \cdot \bar{Y} \cdot \bar{Z}$
  - 2)  $X \cdot Y \cdot Z$
  - 3)  $X + Y + Z$
  - 4)  $\bar{X} + \bar{Y} + \bar{Z}$
- 4) первое выражение,  $\bar{X} \cdot \bar{Y} \cdot \bar{Z}$ , равно 1 только при  $X = Y = Z = 0$ , поэтому это неверный ответ (первая строка таблицы не подходит);
- 5) второе выражение,  $X \cdot Y \cdot Z$ , равно 1 только при  $X = Y = Z = 1$ , поэтому это неверный ответ (первая и вторая строки таблицы не подходит);
- 6) третье выражение,  $X + Y + Z$ , равно нулю при  $X = Y = Z = 0$ , поэтому это неверный ответ (вторая строка таблицы не подходит);
- 7) наконец, четвертое выражение,  $\bar{X} + \bar{Y} + \bar{Z}$  равно нулю только тогда, когда  $X = Y = Z = 1$ , а в остальных случаях равно 1, что совпадает с приведенной частью таблицы истинности;
- 8) таким образом, правильный ответ – 4 ; частичная таблица истинности для всех выражений имеет следующий вид:

$X$	$Y$	$Z$	$F$	$\bar{X} \cdot \bar{Y} \cdot \bar{Z}$	$X \cdot Y \cdot Z$	$X + Y + Z$	$\bar{X} + \bar{Y} + \bar{Z}$
1	0	0	1	0 $\times$	0 $\times$	1	1
0	0	0	1	–	–	0 $\times$	1
1	1	1	0	–	–	–	0

(крестик показывает, что значение функции не совпадает с  $F$ , а знак «–» означает, что вычислять оставшиеся значения не обязательно).

## 7. Электронные таблицы

Адрес ячейки в электронных таблицах состоит из имени столбца и следующего за ним номера строки, например, C15.

Формулы в электронных таблицах начинаются знаком = («равно»)

Знаки +, –, \*, / и ^ в формулах означают соответственно сложение, вычитание, умножение, деление и возвведение в степень.

Запись B2:C4 означает диапазон, то есть, все ячейки внутри прямоугольника, ограниченного ячейками B2 и C4:

	A	B	C	D
1				
2				
3				
4				
5				
6				

Например, по формуле =СУММ(В2:С4) вычисляется сумма значений ячеек В2, В3, В4, С2, С3 и С4.

В заданиях могут использоваться стандартные функции СЧЕТ (количество непустых ячеек), СУММ (сумма), СРЗНАЧ (среднее значение), МИН (минимальное значение), МАКС (максимальное значение).

Функция СРЗНАЧ при вычислении среднего арифметического не учитывает пустые ячейки и ячейки, заполненные текстом; например, после ввода формулы в С2 появится значение 2 (ячейка А2 – пустая):

	A	B	C
1	1	2	
2		3	
3			=СРЗНАЧ(А1:В2)

Функция СЧЕТ(А1:В2) в этом случае выдаст значение 3 (а не 4).

Адреса ячеек (или ссылки на ячейки) бывают относительные, абсолютные и смешанные, вся разница между ними проявляется при копировании формулы в другую ячейку:

- в *абсолютных* адресах перед именем столбца и перед номером строки ставится знак доллара \$, такие адреса не изменяются при копировании; вот что будет, если формулу =\$B\$2+\$C\$3 скопировать из D5 во все соседние ячейки

	C	D	E
4	=B\$2+\$C\$3	=B\$2+\$C\$3	=B\$2+\$C\$3
5	=B\$2+\$C\$3	=B\$2+\$C\$3	=B\$2+\$C\$3
6	=B\$2+\$C\$3	=B\$2+\$C\$3	=B\$2+\$C\$3

Знак \$ как бы «фиксирует» значение: в абсолютных адресах и имя столбца, и номер строки зафиксированы;

- в *относительных* адресах знаков доллара нет, такие адреса при копировании изменяются: номер столбца (строки) изменяется на столько, на сколько отличается номер столбца (строки), где оказалась скопированная формула, от номера столбца (строки) исходной ячейки; вот что будет, если формулу =B2+C3 (в ней оба адреса – относительные) скопировать из D5 во все соседние ячейки:

	C	D	E
4	=A1+B2	=B1+C2	=C1+D2
5	=A2+B3	=B2+C3	=C2+D3
6	=A3+B4	=B3+C4	=C3+D4

- в смешанных адресах часть адреса (строка или столбец) – абсолютная, она «закреплена» знаком \$, а вторая часть – относительная; относительная часть изменится при копировании так же, как и для относительной ссылки:

	C	D	E
4	=\$B1+B\$3	=\$B1+C\$3	=\$B1+D\$3
5	=\$B2+B\$3	=\$B2+C\$3	=\$B2+D\$3
6	=\$B3+B\$3	=\$B3+C\$3	=\$B3+D\$3
7			

### Пример задания

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

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

#### Решение:

- функция **СУММ(B1:B2)** считает сумму значений ячеек B1 и B2, поэтому  $B1 + B2 = 5$
- функция **СРЗНАЧ(B1:B3)** считает среднее арифметическое диапазона B1:B3
- в диапазон **B1:B3** входят три ячейки; предполагаем, что все они содержат числовые данные, тогда среднее арифметическое – это сумма их значений, деленная на 3; таким образом  $B1 + B2 + B3 = 3 \cdot 3 = 9$
- поскольку  $B1 + B2 = 5$ , сразу получаем  $B3 = 9 - 5 = 4$
- таким образом, правильный ответ – 4.

### Пример задания

Дан фрагмент электронной таблицы:

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

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

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

#### Решение:

- в формуле, которая находится в ячейке C1, используются два адреса: A1 и B\$1, адрес A1 – относительный, он может изменяться полностью (и строка, и столбец), адрес B\$1 – смешанный, в нем номер строки

«записан» знаком доллара, а имя столбца – нет, поэтому при копировании может измениться только имя столбца

2) при копировании из C1 в C2 столбец не изменяется, а номер строки увеличивается на 1, поэтому в C2 получим формулу **=A2+B\$1** (здесь учтено, что у второго адреса номер строки «записан»)

6) сумма ячеек A2 и B1 равна  $30 + 20 = 50$ , поэтому правильный ответ – 2.

## 8. Составление запросов для поисковых систем с использованием логических выражений.

Для решения задач такого типа необходимо знать таблицы истинности логических операций «И», «ИЛИ», «НЕ» (см. тему №6).

Если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ».

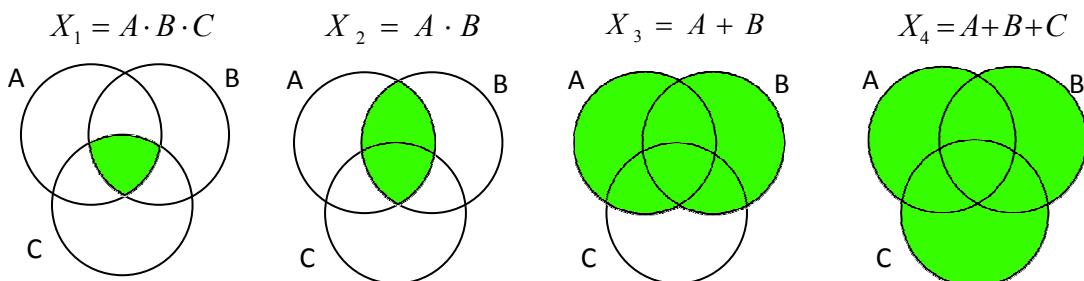
Ввод какого-то слова (скажем, яблоко) в запросе поисковой системы означает, что пользователь ищет Web-страницы, на которых встречается это слово.

Операция «И» всегда ограничивает поиск, то есть, в ответ на запрос **яблоко И груша** поисковый сервер выдаст меньше страниц, чем на запрос **яблоко**, потому что будет искать страницы, на которых есть оба этих слова одновременно.

Операция «ИЛИ» всегда **расширяет** поиск, то есть, в ответ на запрос **яблоко ИЛИ груша** поисковый сервер выдаст **больше** страниц, чем на запрос **яблоко**, потому что будет искать страницы, на которых есть хотя бы одно из этих слов (или оба одновременно).

Для наглядного представления результатов логических операций над множествами удобно использовать диаграммы (круги) Эйлера. При анализе поискового запроса множество – это отдельное слово.

Для запроса, состоящего из трех слов (множеств A, B и C) диаграммы Эйлера выглядят следующим образом:



### Пример задания

В таблице приведены запросы и количество страниц, которые нашел поисковый сервер по этим запросам в некотором сегменте Интернета:

Запрос	Количество страниц (тыс.)
Динамо & Рубин	320
Спартак & Рубин	280
(Динамо   Спартак) & Рубин	430

Сколько страниц (в тысячах) будет найдено по запросу

**Рубин & Динамо & Спартак**

**Решение (вариант 1, круги Эйлера, полная диаграмма):**

1) в этой задаче неполные данные, так как они не позволяют определить размеры всех областей; однако их хватает для того, чтобы ответить на поставленный вопрос

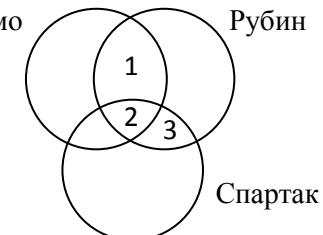
2) обозначим области, которые соответствуют каждому запросу

Запрос	Области	Количество страниц (тыс.)
Динамо & Рубин	1+2	320
Спартак & Рубин	2+3	280
(Динамо   Спартак) & Рубин	1+2+3	430
<b>Рубин &amp; Динамо &amp; Спартак</b>	<b>2</b>	?

3) из таблицы следует, что в суммарный результат первых двух запросов область 2 входит дважды ( $1 + 2 + 2 + 3$ ), поэтому, сравнивая этот результат с третьим запросом ( $1 + 2 + 3$ ), сразу находим результат четвертого:

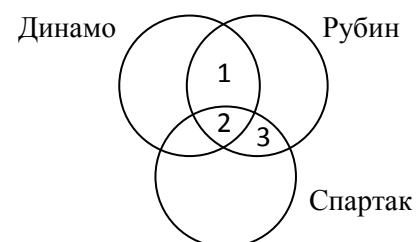
$$N_2 = (320 + 280) - 430 = 170$$

4) таким образом, ответ – 170.



**Решение (вариант 2, круги Эйлера, неполная диаграмма):**

1) заметим, что в этой задаче все запросы (в том числе и тот, результат которого нужно найти, имеют вид **X & Рубин**



2) поэтому часть «& Рубин» в каждом из запросов можно просто отбросить, тогда останется только две области:

<i>Запрос</i>	<i>Количество стран</i> (тыс.)
Динамо-1	320
Спартак-1	280
Динамо-1   Спартак-1	430

здесь добавление «-1» в имени области обозначает «пересечение с областью Рубин»

- 3) требуется найти размер области «Динамо-1 & Спартак-1»
- 4) для диаграммы с двумя областями можно использовать общую формулу

$$N_{A|B} = N_A + N_B - N_{A \& B}$$

5) из которой следует

$$N_{A \& B} = N_A + N_B - N_{A|B}$$

6) в данном случае получаем

$$N_{A \& B} = (320 + 280) - 430 = 170$$

7) таким образом, ответ – 170.

## 9. Программирование базовых алгоритмических структур

**ВАЖНО!** Программы к заданиям 9 и 10 можно писать на любом языке программирования. Ниже приводится теория и рассматриваются задачи, решенные с помощью языка программирования РАВС.

### *Краткий обзор языка программирования РАВС*

#### *Структура программы*

##### *1. Заголовок*

*Program <имя>;*

##### *2. Раздел описаний*

*Uses <список библиотек>;  
type <список типов>;  
const <список констант>;  
var <список переменных>;  
procedure <имя>;  
function <имя>:<тип>;*

##### *3. Раздел операторов*

*BEGIN*

...

*END.*

## Типы данных

Любая переменная, используемая в программе должна быть описана в разделе описания переменных, т.е. после имени переменной через двоеточие должен стоять ее тип. Однотипные переменные можно перечислять через запятую, наборы разнотипных переменных отделяются друг от друга точкой с запятой.

*Integer* – целый тип данных. Область значений от -32768 до 32767.

*Real* – вещественный тип данных. Область значений от -1038 до 1038.

*Byte* – порядковый тип данных. Область значений от 0 до 255. (Пр., 0, 2, 52, 150).

*Char* – символьный тип данных. Область значений все символы.

*Boolean* – логический тип данных. Область значений TRUE или FALSE.

Например:

*Var*

*a: real;*

*b,c:integer;*

## Операторы ввода

*read (<список параметров>);*

*readln (<список параметров>);*

<список параметров> - это набор считываемых переменных целого, вещественного, символьного и строкового типов. Если переменных несколько, они разделяются запятой.

Например:

*read (a);*

*read (c,d,e);*

## Операторы вывода

*write (<список вывода>);*

*writeln (<список вывода>);*

Список вывода может состоять из констант, переменных и выражений целого, вещественного, логического, символьного и строкового типов.

Например:

*writeln ('Ввести a и b');*

*writeln (a); //10*

```
writeln ('a=' ,a); //a=10
writeln (a, 'km'); //a km
```

### Оператор присваивания

“:=” – оператор присваивания.

В его левой части указывается имя переменной, а в правой – выражение такого же типа, что и переменная. Значение выражения вычисляется и присваивается переменной.

Например:

```
a:=a+b;
```

### Условный оператор (развилка)

Развилка служит для выбора одного из двух возможных действий в зависимости от истинности условия.

Структура оператора развилки имеет следующий вид:

```
IF <условие>
  THEN
    <оператор 1>
  ELSE
    <оператор 2>;
```

Например, если  $a>b$  требуется найти  $R=a-b$ , иначе найти  $R=b-a$ .

Фрагмент программы:

```
IF a>b
  THEN
    R:=a-b
  ELSE
    R:=b-a;
```

### Составной оператор (операторные скобки)

В том случае, когда по ветке развилки или цикла стоит два и более операторов необходимо ставить операторные скобки (*begin ... end*).

Например, дано два числа  $a$  и  $b$ , если  $a>b$  найти и вывести значение  $S=a+b$ , иначе  $P=a*b$ .

```
If a>b then
  begin
    S:=a+b;
    writeln ('S=',S);
  end
else
  begin
```

```

P:=a*b;
writeln ('P=',P);
end;

```

## Циклы

Циклы предназначены для программирования повторяющихся фрагментов алгоритмов. В информатике существует три вида циклов.

1. **Цикл с предусловием** – операторы в цикле (тело цикла) повторяются, пока условие окончания цикла истинно. Структура оператора имеет следующий вид:

```

While <условие> do
  <оператор>;

```

Для примера приведена программа вычисления значения функции  $y=x^2$  при изменении аргумента на интервале  $[-3, 3]$  с шагом 0,2:

```

Program PR1;
Var
  x,y : real;
Begin
  x:=-3;
  While x<=3 do
    Begin
      y:=sqr(x);
      Writeln ('x=',x:6:2);
      Writeln ('y=',y:6:2);
      x:=x+0.2;
    end;
  End.

```

2. **Цикл с постусловием** – операторы в цикле (тело цикла) повторяются, до тех пор, пока условие окончания цикла не станет ложным. Структура оператора имеет следующий вид:

```

Repeat
  <оператор>;
until <условие>;

```

Для примера приведена программа вычисления значения функции  $y=3x$  при изменении аргумента на интервале  $[-5, 5]$  с шагом 0,5:

```

Program PR2;
Var

```

```

x,y : real;
Begin
  x:=-5;
  repeat
    y:=3*x;
    Writeln ('x=',x:6:2);
    Writeln ('y=',y:6:2);
    x:=x+0.5;
  Until x>5;
End.

```

3. **Цикл с параметром** – используется в тех случаях, когда заранее известно число повторений цикла. Структура оператора имеет следующий вид:

*For <переменная>:=<нач. значение> to <конечное значение> do  
<оператор>;*

или

*For <переменная>:=<нач. значение> downto <конечное значение>  
do  
<оператор>;*

Для примера приведена программа вычисления суммы n последовательно введенных чисел.

```

Program PR3;
Var
  n,i: integer;
  a,s : real;
Begin
  Writeln ('Ввести n');
  Readln(n);
  S:=0;
  For i:=1 to n do
    Begin
      Writeln ('Ввести a');
      Readln(a);
      S:=S+a;
    End;
  Writeln ('S=',S:6:2);

```

*End.*

### **Пример задания**

Ввести четырёхзначное число, выяснить равны ли первая и последняя цифра в его записи. Если цифры равны и чётны, то разделить их на два. Если цифры не равны и первая цифра больше последней, то поменять их местами. Если цифры не равны и первая цифра меньше последней, то поменять местами вторую и третью цифры.

```
Program Ciferki;  
var  
  X,Y,x1,x2,x3,x4: integer;  
begin  
  cls;  
  writeln('Введите четырёхзначное число');  
  readln(X);  
  //чтобы не потерять исходное число, копируем его в Y  
  Y:=X;  
  x1:=Y div 1000; // первая цифра  
  Y:= Y mod 1000; // отбрасываем первую цифру  
  x2:=Y div 100; // вторая цифра  
  Y:= Y mod 100; // отбрасываем вторую цифру  
  x3:=Y div 10; // третья цифра  
  Y:= Y mod 10; // отбрасываем третью цифру  
  x4:=Y div 1; // четвёртая цифра  
  // проверяем правильно ли мы разложили введённое число  
  Y:= 1000*x1 + 100*x2 + 10*x3 + x4;  
  writeln('на входе было', X, ', на выходе получилось ',Y);  
  writeln('1-я цифра:', x1);  
  writeln('2-я цифра:', x2);  
  writeln('3-я цифра:', x3);  
  writeln('4-я цифра:', x4);  
  if x1=x4 then  
    begin  
      if x1 mod 2=0 then //x4 не проверяем, т.к. x1=x4  
        begin  
          writeln('первая и последняя цифры четны и равны');  
          x1:=x1 div 2;
```

```

x4:=x4 div 2;
end
else
    writeln('1-я и последняя цифры равны, но не четны');
end
else
begin
    writeln('первая и последняя цифры не равны');
    if x1>x4 then
begin
    Y:=x1;
    x1:=x4;
    x4:=Y;
end
else
begin
    Y:=x2;
    x2:=x3;
    x3:=Y;
end
end;
X:= 1000*x1 + 100*x2 + 10*x3 + x4;
writeln('в результате преобразований получено число ', X);
end.

```

## 10. Алгоритмы обработки одномерных массивов

**ВАЖНО!** Программы к заданиям 9 и 10 можно писать на любом языке программирования. Ниже приводится теория и рассматриваются задачи, решенные с помощью языка программирования ПАВС.

### Одномерные массивы

Элемент массива $A$ :	-4	3	2	0	-7	5	2	10	1
Индекс элемента	1	2	3	4	5	6	7	8	9

Массив – это именованный набор однотипных элементов, упорядоченных по индексу.

Индекс – порядковый номер элемента в массиве. Для того, чтобы обратиться к первому элементу массива, например, записать в первую ячейку число 10, необходимо написать  $A[1]:=10$ ,

### ***Объявление массива***

#### ***1 способ***

*Var <имя>: array [<нач. индекс>..<конеч.индекс>] of <тип>;*

Например, описание массива А из 50 целочисленных элементов:

*A: array [1..50] of integer;*

#### ***2 способ***

*Type <имя\_типа>= array [<нач. индекс>..<конеч.индекс>] of <тип>;*

*Var <имя\_массива>:<имя\_типа>;*

Например, описание массива В массив А из 70 вещественных элементов:

*Type Massiv = array [1..70] of real;*

*Var B: Massiv;*

### ***Базовые алгоритмы обработки одномерных массивов***

#### ***Ввод массива***

*Writeln ('Ввести количество элементов');*

*Readln (N);*

*For i:=1 to N do*

*begin*

*Write('A[ ',i, '] =');*

*Readln (A[i]);*

*end;*

#### ***Вывод массива***

*For i:=1 to N do*

*Writeln('A[ ',i, '] =',A[i]);*

#### ***Поиск суммы***

*S:=0;*

*For i:=1 to N do*

*S:=S+A[i];*

*Writeln ('S= ',S);*

#### ***Поиск произведения***

*P:=1;*

*For i:=1 to N do*

*P:=P\*A[i];*

*Writeln ('P= ',P);*

**Поиск количества по условию**

**(на примере поиска количества положительных элементов)**

*K:=0;*

*For i:=1 to N do*

*If A[i]>0 then*

*K:=K+1;*

*Writeln ('K= ',K);*

**Поиск среднего арифметического по условию**

**(на примере среднего арифметического четных элементов)**

*S:=0;*

*K:=0;*

*For i:=1 to N do*

*If A[i] mod 2 = 0 then*

*begin*

*S:=S+A[i];*

*K:=K+1;*

*end;*

*Sa:=S/K;*

*Writeln ('Sa= ', Sa:6:2);*

**Поиск максимального элемента массива и его местоположения**

*Max:=A[1];*

*Imax:=1;*

*For i:=1 to N do*

*If A[i]>Max then*

*begin*

*Max:=A[i];*

*Imax:=i;*

*end;*

**Пример задания**

Дан одномерный массив  $T(N)$ . Найти в массиве минимальный элемент и сумму четных элементов. Затем заменить этой суммой все отрицательные элементы, стоящие после минимального элемента.

```

PROGRAM ZADACHA;
CONST
  N1=50;
TYPE
  MAS1=ARRAY[1..N1] OF INTEGER;
VAR
  T:MAS1;
  IMIN,MIN,S,I,N:INTEGER;
BEGIN
  WRITE('ВВЕДИТЕ РАЗМЕР N=');
  READ(N);
  FOR I:=1 TO N DO
    BEGIN
      WRITE('T[',I,']= ');
      READ(T[I]);
    END;
  MIN:=T[1];
  IMIN:=1;
  FOR I:=2 TO N DO
    IF T[I]<MIN THEN
      BEGIN
        MIN:=T[I];
        IMIN:=I;
      END;
  WRITELN('МИНИМАЛЬНЫЙ ЭЛЕМЕНТ МАССИВА: ',MIN);
  S:=0;
  FOR I:=1 TO N DO
    IF T[I] mod 2=0 THEN
      S:=S+T[I];
  WRITELN('СУММА ЧЕТНЫХ ЭЛЕМЕНТОВ МАССИВА: ',S);
  IF IMIN<N THEN
    BEGIN
      FOR I:=IMIN+1 TO N DO
        IF T[I]<0 THEN
          T[I]:=S;
      FOR I:=1 TO N DO
        WRITELN('T[',I,']=',T[I]:4);
    END;
END.

```

```
    END
    ELSE
        WRITE('МИНИМАЛЬНЫЙ ЭЛЕМЕНТ ПОСЛЕДНИЙ');
        READLN;
    END.
```

### **Список рекомендуемой литературы**

1. Павлова, Е.С. Алгоритмы обработки числовых массивов: учеб. пособие / О. А. Авдеюк, Е. С. Павлова, В. С. Поляков; ВолгГТУ. - Волгоград, 2017. - 72 с.
2. Малясова, С.В. Информатика и ИКТ: Пособие для подготовки к ЕГЭ / С.В. Малясова; Под ред. Цветковой М.С.. - М.: Academia, 2018. - 637 с.
3. Плотникова, Н.Г. Информатика и информационно-коммуникационные технологии (ИКТ): Учебное пособие / Н.Г. Плотникова. - М.: Риор, 2018. - 132 с.
4. Демидович, Н.Б. Программирование и ЭВМ. Учебное пособие по факультативному курсу для учащихся 9, 10 классов / Н.Б. Демидович, В.М. Монахов. - М.: Просвещение, 2014. - 240 с.
5. Демина, О.А. Экзамен по информатике / О.А. Демина. - М.: Приор, 2012. - 176 с.

Учебное издание

Елена Станиславна **Павлова**  
Ирина Геннадьевна **Лемешкина**  
Игорь Леонидович **Гоник**  
Евгений Владимирович **Кузьмин**  
Юрий Васильевич **Левин**

## ИНФОРМАТИКА

*Учебное пособие для поступающих в ВолгГТУ*

Редактор *E. B. Кравцова*

Темпплан 2021 г. (учебники и учебные пособия). Поз. № 155.  
Подписано в печать 26.10.2021 г. Формат 60×84 1/16. Бумага газетная.  
Гарнитура Times. Печать офсетная. Усл. печ. л. 1,63. Уч.-изд. л. 1,62.  
Тираж 50 экз. Заказ .

Волгоградский государственный технический университет  
400005, г. Волгоград, пр. Ленина, 28, корп. 1.

Отпечатано в типографии ИУНЛ ВолгГТУ  
400005, г. Волгоград, пр. Ленина, 28, корп. 7.