

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»

Т. В. Ромашкина, Н. И. Миндоров

# ИНФОРМАТИКА И ОСНОВЫ ПРОГРАММИРОВАНИЯ

*Допущено методическим советом  
Пермского государственного национального  
исследовательского университета в качестве учебного пособия  
для студентов, обучающихся по направлениям подготовки бакалавров  
«Математика» и «Механика и математическое моделирование»*



Пермь 2018

УДК 004  
ББК 32.973-018  
Р698

**Ромашкина Т. В., Миндоров Н. И.**

**Р698** Информатика и основы программирования [Электронный ресурс]: учеб. пособие / Т. В. Ромашкина, Н. И. Миндоров; Перм. гос. нац. исслед. ун-т. Электрон. дан. – Пермь, 2018. – 2,5 Мб; 111 с. – Режим доступа: [www.psu.ru/files/docs/science/books/uchebnie-posobiya/romashkina-mindorov-informatika-i-osnovy-programmirovaniya.pdf](http://www.psu.ru/files/docs/science/books/uchebnie-posobiya/romashkina-mindorov-informatika-i-osnovy-programmirovaniya.pdf). – Загл. с экрана.  
ISBN 978-5-7944-3182-7

Учебное пособие состоит из двух частей. В первой рассматриваются отдельные вопросы общей информатики, системы счисления, машинное представление числа, измерение информации, кодирование информации, работа процессора. Во второй части – основы объектно-ориентированного программирования с использованием языка программирования С#.

**УДК 004  
ББК 32.973-018**

*Печатается по решению ученого совета  
механико-математического факультета  
Пермского государственного национального исследовательского университета*

*Рецензенты:* кафедра информатики и вычислительной техники Пермского государственного гуманитарно-педагогического университета (зав. кафедрой к. пед. н., доцент **А. П. Шестаков**), д. ф.-м. н., профессор кафедры математики и социально-гуманитарных дисциплин Пермского филиала РАНХиГС **К. Г. Шварц**

ISBN 978-5-7944-3182-7

© ПГНИУ, 2018  
© Ромашкина Т. В., Миндоров Н. И., 2018

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ</b> .....	5
<b>СИСТЕМЫ СЧИСЛЕНИЯ</b> .....	6
ТРАДИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ.....	6
Задания для выполнения в аудитории .....	13
Задания для самостоятельной работы студента (СРС) .....	14
СМЕШАННЫЕ И НЕТРАДИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ .....	16
Задания для выполнения в аудитории .....	20
Задания для самостоятельной работы студента (СРС) .....	21
ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ .....	22
Приложение 1.....	23
Приложение 2.....	24
<b>МАШИННОЕ ПРЕДСТАВЛЕНИЕ ЧИСЕЛ</b> .....	25
ПРЕДСТАВЛЕНИЕ ЦЕЛЫХ ЧИСЕЛ В ПАМЯТИ КОМПЬЮТЕРА.....	25
ПРЕДСТАВЛЕНИЕ ВЕЩЕСТВЕННЫХ ЧИСЕЛ В ПАМЯТИ КОМПЬЮТЕРА.....	28
Задания для выполнения в аудитории .....	31
Задания для самостоятельной работы студента (СРС) .....	32
ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ .....	34
<b>ИЗМЕРЕНИЕ ИНФОРМАЦИИ</b> .....	35
СОБЫТИЯ РАВНОВЕРОЯТНЫЕ .....	35
Задания для выполнения в аудитории .....	39
Задания для самостоятельной работы студента (СРС) .....	40
СОБЫТИЯ НЕРАВНОВЕРОЯТНЫЕ .....	42
Задания для выполнения в аудитории .....	46
Задания для самостоятельной работы студента (СРС) .....	47
ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ .....	49
<b>КОДИРОВАНИЕ ИНФОРМАЦИИ</b> .....	50
КОД ШЕННОНА – ФАНО .....	51
Задания для выполнения в аудитории .....	54
Задания для самостоятельной работы студента (СРС) .....	55
КОД ХЕММИНГА.....	60

Задания для выполнения в аудитории .....	63
Задания для самостоятельной работы студента (СРС) .....	65
ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ .....	67
<b>РАБОТА ПРОЦЕССОРА</b> .....	68
КОМАНДЫ ПРОЦЕССОРА .....	69
Задания для выполнения в аудитории .....	76
Задания для самостоятельной работы студента (СРС) .....	78
КОСВЕННЫЙ СПОСОБ АДРЕСАЦИИ .....	80
Задания для выполнения в аудитории .....	82
Задания для самостоятельной работы студента (СРС) .....	84
ОРГАНИЗАЦИЯ ПЕРЕХОДОВ В ПРОГРАММАХ. ВЕТВЛЕНИЯ И ЦИКЛИЧЕСКИЕ КОНСТРУКЦИИ.....	86
Задания для выполнения в аудитории .....	88
Задания для самостоятельной работы студента (СРС) .....	89
ОРГАНИЗАЦИЯ ПЕРЕХОДОВ. СТЕК. ПОДПРОГРАММЫ.....	93
Задания для выполнения в аудитории .....	96
Задания для самостоятельной работы студента (СРС) .....	98
ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ .....	99
Приложение 1 .....	100
Приложение 2.....	109
Приложение 3.....	110

## ВВЕДЕНИЕ

---

Учебно-практическое пособие по дисциплине «Информатика и основы программирования» предназначено для изучения указанной дисциплины студентами первого курса направлений «Математика» и «Механика и математическое моделирование».

Пособие состоит из двух частей.

В первой части рассматриваются отдельные вопросы общей информатики, системы счисления, машинное представление чисел, измерение информации, кодирование информации, работа процессора.

Во второй части – основы объектно-ориентированного программирования с использованием языка программирования C#.

Структура пособия представлена следующим образом:

- Теоретические сведения по изучаемым разделам с примерами решения задач. Приведенные примеры имеют подробное описание процесса решения. По окончании рассмотрения примеров имеются гиперссылки для перехода к выполнению заданий по данной теме.
- Задания по изученному теоретическому материалу:
  - *задания для выполнения в аудитории.* Для самоконтроля и правильности выполнения аудиторных заданий студентам предоставляется возможность воспользоваться активными кнопками «Указания к выполнению» и «Показать ответ»;
  - *задания для самостоятельной работы студента (СРС).* Данные задания предназначены как для закрепления изученного материала, так и для самостоятельного изучения некоторых тем дисциплины. Материал, предназначенный для самостоятельного освоения, снабжен пошаговыми инструкциями (алгоритмом работы).
- Информационные источники по каждому из разделов.
- Приложения с необходимой справочной информацией.

Для информационной поддержки самостоятельной работы студентов в пособии используются:

- электронный ресурс Online Binary-Decimal Converter (Интерактивный двоично-десятичный конвертор с поддержкой стандарта IEEE754);
- разработки преподавателей ПГНИУ (Т.С. Белозерова, Н.И. Миндоров, С.В. Русаков) «Интегрированная среда для изучения арифметических основ работы ЭВМ», «Измерение информации».

# СИСТЕМЫ СЧИСЛЕНИЯ

## ТРАДИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

*Традиционная* система счисления – это система счисления, в которой базис состоит из членов геометрической прогрессии.

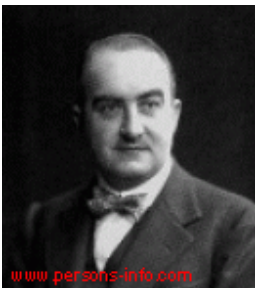
Знаменатель  $P$  геометрической прогрессии называется *основанием* системы счисления:

$$\dots, P^{-2}, P^{-1}, P^0, P^1, P^2, P^3, \dots, P^n \dots$$

Традиционные позиционные системы счисления с основанием  $P$  называются  **$P$ -ичными** системами счисления.

К традиционным позиционным системам счисления относятся: десятичная, двоичная, восьмеричная, шестнадцатеричная и др.

### Представление чисел в $P$ -ичной СС



Горнер Уильям Джордж (1786 – 1837)

Любое вещественное число  $X$  с основанием  $P$  в общем виде может быть представлено с помощью общей схемы Горнера (развернутая форма записи числа):

$$X_p = a_n P^n + a_{n-1} P^{n-1} + a_{n-2} P^{n-2} + \dots + a_1 P^1 + a_0 P^0 + b_{-1} P^{-1} + b_{-2} P^{-2} + \dots + b_{-k} P^{-k} + \dots, \text{ (формула 1)}$$

где

$X$  – любое вещественное число;

$P$  – основание данной позиционной системы счисления;

$a, b$  – цифры данной позиционной системы счисления;

$n$  – количество разрядов целой части числа  $X$ ,  $k$  – количество разрядов дробной части числа  $X$ .

**Пример 1.** Свернутая и развернутая формы записи числа (десятичная СС).

Свернутая форма записи числа:  $4312,321_{10}$

$$\text{Развернутая форма записи числа: } 4312,321_{10} = 4 \cdot 10^3 + 3 \cdot 10^2 + 1 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} + 2 \cdot 10^{-2} + 1 \cdot 10^{-3}$$

**Пример 2.** Свернутая и развернутая формы записи числа (двоичная СС).

Свернутая форма записи числа:  $1001,001_2$

Развернутая форма записи числа:

$$1001,001_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$1001,001_2 = 1 \cdot 10^3 + 0 \cdot 10^2 + 0 \cdot 10^1 + 0 \cdot 10^0 + 1 \cdot 10^0 + 0 \cdot 10^{-1} + 0 \cdot 10^{-2} + 1 \cdot 10^{-3}$$

в данном примере  $1 \cdot 10^3$ ;  $10_2 = 2_{10}$ .

## 1. Перевод целых чисел из одной позиционной системы счисления в другую

### 1.1. Перевод из десятичной системы счисления в любую другую систему счисления

Перевод из десятичной системы счисления в любую другую осуществляется делением числа на основание системы счисления, с выделением остатка от деления.

**Пример 3.** Переведите число  $14_{10}$  в двоичную систему счисления:  $14_{10} = ?_2$

**Решение:**

$$\begin{array}{r|l}
 14 & 2 \\
 \hline
 14 & 7 \\
 \hline
 0 & 6 \\
 & \hline
 & 3 \\
 & \hline
 & 2 \\
 & \hline
 & 1 \\
 & \hline
 & 1
 \end{array}$$

←

Ответ:  $14_{10} = 1110_2$

**Пример 4.** Переведите число  $14_{10}$  в систему счисления с основанием 4:  $14_{10} = ?_4$

**Решение:**

Алфавит системы счисления с основанием «4»: 0; 1; 2; 3.

$$\begin{array}{r|l}
 14 & 4 \\
 \hline
 12 & 3 \\
 \hline
 2 &
 \end{array}$$

←

Ответ:  $14_{10} = 32_4$

**Пример 5.** Переведите число  $14_{10}$  в восьмеричную систему счисления:  $14_{10} = ?_8$

**Решение:**

Алфавит системы счисления с основанием «8»: 0; 1; 2; 3; 4; 5; 6; 7.

$$\begin{array}{r|l}
 14 & 8 \\
 \hline
 8 & 1 \\
 \hline
 6 &
 \end{array}$$

←

Ответ:  $14_{10} = 16_8$

**Пример 6.** Переведите число  $14_{10}$  в шестнадцатеричную систему счисления:  $14_{10} = ?_{16}$

**Решение:**

$14 < 16$  (основание системы счисления; 14 входит в алфавит системы счисления с основанием 16, см. таблицу 1)

Ответ:  $14_{10} = E_{16}$

Таблица 1

Система счисления										
«10»	«2»	«4»	«8»	«16»		«10»	«2»	«4»	«8»	«16»
0	0	0	0	0		9	1001	21	11	9
1	1	1	1	1		10	1010	22	12	A
2	10	2	2	2		11	1011	23	13	B
3	11	3	3	3		12	1100	30	14	C
4	100	10	4	4		13	1101	31	15	D
5	101	11	5	5		14	1110	32	16	E
6	110	12	6	6		15	1111	33	17	F
7	111	13	7	7		16	10000	40	20	10
8	1000	20	10	8						

**Пример 7.** Переведите число  $44_{10}$  в шестнадцатеричную систему счисления:  $44_{10} = ?_{16}$

**Решение:**

$$\begin{array}{r|l} 44 & 16 \\ -32 & 2 \\ \hline 12 & \end{array}$$

$12_{10} = C_{16}$  (см. таблицу1)

**Ответ:**  $44_{10} = 2C_{16}$

### 1.2. Перевод из любой другой системы счисления в десятичную систему счисления

Перевод из любой системы счисления в десятичную систему счисления осуществляется с использованием развернутой формы записи числа (см. Пример 1, Пример 2).

**Пример 8.** Переведите в десятичную систему счисления следующие числа:

- 1)  $101101011_2 = ?_{10}$     2)  $702_8 = ?_{10}$     3)  $A0C1_{16} = ?_{10}$     4)  $213_5 = ?_{10}$

**Решение:**

1)  $101101011_2 = ?_{10}$

8 7 6 5 4 3 2 1 0

$$101101011_2 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6 + 0 \cdot 2^7 + 1 \cdot 2^8 = 363_{10}$$

2)  $702_8 = ?_{10}$

2 1 0

$$702_8 = 2 \cdot 8^0 + 0 \cdot 8^1 + 7 \cdot 8^2 = 450_{10}$$

3)  $A0C1_{16} = ?_{10}$

3 2 1 0

$$A0C1_{16} = 1 \cdot 16^0 + C \cdot 16^1 + 0 \cdot 16^2 + A \cdot 16^3 = 1 + 12 \cdot 16 + 0 + 10 \cdot 4096 = 41153_{10}$$

4)  $213_5 = ?_{10}$

2 1 0

$$213_5 = 3 \cdot 5^0 + 1 \cdot 5^1 + 2 \cdot 5^2 = 83_{10}$$



## 2. Перевод вещественных чисел из одной позиционной системы счисления в другую

### 2.1. Перевод из десятичной системы счисления в любую другую систему счисления

Перевод из десятичной системы счисления в любую другую:

- 1) умножение исходной дроби на основание системы счисления;
- 2) выделение целой части от результата умножения (цифра получаемой дроби);
- 3) оставшаяся дробная часть вновь умножается на основание системы счисления.

Умножение продолжается до тех пор, пока не будет получен нулевой результат, или не будет достигнуто требуемое количество цифр в получаемой дроби.

**Пример 9.** Переведите число  $0,43_{10}$  в следующие системы счисления:

- 1) двоичную, получив шесть значащих цифр после запятой:  $0,43_{10} = ?_2$
- 2) восьмеричную, получив пять значащих цифр после запятой:  $0,43_{10} = ?_8$
- 3) шестнадцатеричную, получив четыре значащих цифр после запятой:  $0,43_{10} = ?_{16}$

**Решение:**

- 1)  $0,43_{10} = ?_2$

$$\begin{array}{r} 0,43 \\ * 2 \\ \hline 0,86 \rightarrow 0,86 \\ * 2 \\ \hline 1,72 \rightarrow 0,72 \\ * 2 \\ \hline 1,44 \rightarrow 0,44 \\ * 2 \\ \hline 0,88 \rightarrow 0,88 \\ * 2 \\ \hline 1,76 \rightarrow 0,76 \\ * 2 \\ \hline 1,52 \dots \end{array}$$

Процесс умножения прекращен, так как получено требуемое количество знаков после запятой.

Ответ:  $0,43_{10} = 0,011011_2$

- 2)  $0,43_{10} = ?_8$

$$\begin{array}{r} 0,43 \\ * 8 \\ \hline 3,44 \rightarrow 0,44 \\ * 8 \\ \hline 3,52 \rightarrow 0,52 \\ * 8 \\ \hline 4,16 \rightarrow 0,16 \\ * 8 \\ \hline 1,28 \rightarrow 0,28 \\ * 8 \\ \hline 2,24 \dots \end{array}$$

Процесс умножения прекращен, так как получено требуемое количество знаков после запятой.

Ответ:  $0,43_{10} = 0,33412_8$

3)  $0,43_{10} = ?_{16}$

$$\begin{array}{r}
 0,43 \\
 * \quad 16 \\
 \hline
 6,88 \longrightarrow 0,88 \\
 \quad * \quad 16 \\
 \quad \hline
 \quad 14,08 \longrightarrow 0,08 \\
 \quad \quad * \quad 16 \\
 \quad \quad \hline
 \quad \quad 1,28 \longrightarrow 0,28 \\
 \quad \quad \quad * \quad 16 \\
 \quad \quad \quad \hline
 \quad \quad \quad 4,48 \longrightarrow \dots
 \end{array}$$

Процесс умножения прекращен, так как получено требуемое количество знаков после запятой.

$0,43_{10} = 0,61414_{16} = 0,6E14_{16}$

Ответ:  $0,43_{10} = 0,6E14_{16}$

**Пример 10.** Переведите число  $0,(6)_{10}$  в двоичную систему счисления.  $0,(6)_{10} = ?_2$

**Решение:**

1. Период дроби умножается на основание 2:

$$6 * 2 = 12$$

2. Проверяется на равенство количество цифр в периоде и в полученном произведении:

$(6)$  – в периоде одна цифра, а в произведении две цифры **1** и **2**. Цифра **1** считается "лишней".

3. Складываются цифры полученного произведения:  $12 \rightarrow 1+2=3$

4. Вновь проверяется на равенство количество цифр в периоде и в полученном действии:  $1+2=3 \rightarrow$  "3" одна цифра. Следовательно, период произведения = 3. **(3)**

5. Из лишних цифр формируется целая часть числа в требуемой системе счисления (если "лишних" чисел несколько, то они складываются). В данном примере "лишняя" только цифра **1**. Следовательно, результат умножения: **1, (3)**.

6.

$$\begin{array}{r}
 0,(6) \\
 * \quad 2 \\
 \hline
 1,(3) \longrightarrow 0,(3) \\
 \quad * \quad 2 \\
 \quad \hline
 \quad 0,(6) \longrightarrow 0,(6) \\
 \quad \quad * \quad 2 \\
 \quad \quad \hline
 \quad \quad 1,(3) \dots
 \end{array}$$

Присутствует повторение действий (период **(10)<sub>2</sub>**)

Следовательно,  $0,(6)_{10} = 0,(10)_2$



1) Перевод целой части

$$\begin{array}{r} 25 \mid 5 \\ - 25 \mid 5 \mid 5 \\ \hline 0 \quad 5 \mid 1 \\ \hline 0 \end{array}$$

$$25_{10} = 100_5$$

2) Перевод дробной части:

$$\begin{array}{r} 0,4 \\ * 5 \\ \hline 2,0 \end{array}$$

$$0,4_{10} = 0,2_5$$

3) Сложить результат перевода:  $25_{10} + 0,4_{10} = 100_5 + 0,2_5 = 100,02_5$

**Пример 14.** Переведите в систему счисления с основанием 2 число  $0,(6)_{10}$

**Решение:**

1. Периодическая дробь представляется как обыкновенная дробь: в числителе располагается период, в знаменателе – 10 в степени, соответствующей количеству цифр периода, уменьшенным на единицу.

2. Целочисленные числитель и знаменатель переводятся в  $p$ -ичную систему.

3. Делится числитель на знаменатель – результат  $p$ -ичная дробь.

$$0,(6)_{10} = ?_2$$

$$0,(6) = \frac{6}{10^1 - 1} = \frac{6}{9} = \frac{2}{3} = \frac{10_2}{11_2} = 0,(10)_2$$

### 3.2. Перевод из любой другой системы счисления в десятичную систему счисления

**Пример 15.** Переведите в десятичную систему счисления число  $1110,011011_2$

**Решение:**

$$1110,011011_2 = ?_{10}$$

$$3 \ 2 \ 1 \ 0 \ -1 \ -2 \ -3 \ -4 \ -5 \ -6$$

$$1110,011011_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} = 14,43_{10}$$

**Пример 16.** Переведите в десятичную систему счисления число  $0,(1011)_2$

**Решение:**

$$0,(1011)_2 = ?_{10}$$

Исходная дробь представляется в виде обыкновенной дроби:

- в числителе записывается период в развернутой форме;
- в знаменателе –  $p$  (основание системы счисления) в степени, равной количеству знаков в периоде, уменьшенной на единицу.

$$0,(1011)_2 = \frac{1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0}{2^4 - 1} = \frac{11}{15} = 0,7(3)_{10}$$

**Пример 17.** Переведите в десятичную систему счисления число  $0,10(21)_3$

**Решение:**

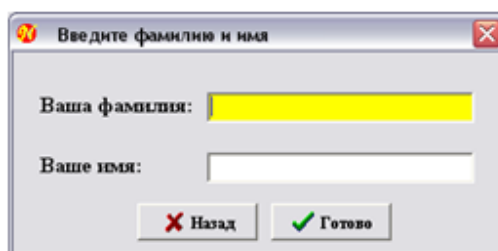
$$0,10(21)_3 = ?_{10}$$

$$0,10(21)_3 = 0,1_3 + 0,00(21)_3 = 3^{-1} + 3^{-2} \cdot \frac{2 \cdot 3^2 + 1 \cdot 3^0}{3^2 - 1} = \frac{1}{3} + \frac{19}{9 \cdot 8} = 0,597(2)_{10}$$

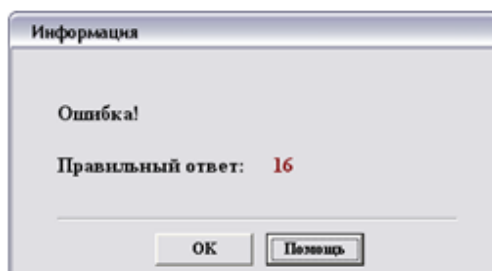
## Задание №1. Работа с Numbers. Целые числа

### 1.1. Выполнение заданий по теме «Система счисления - Целые числа», Тренировка:

- Запустите Numbers.
- В появившемся диалоговом окне выберите пункт меню «Старт».
- Введите свои фамилию, имя и щелкните «Готово»:



- Выберите тему «Система счисления – Целые числа» (при необходимости Вы можете воспользоваться «Справкой по теме»).
- Выберите «Готово».
- Выберите тип занятия «Тренировка» – «ОК».
- Обратите внимание: при неправильном ответе Вы можете воспользоваться помощью (кнопка «Помощь») и «увидеть» процесс получения верного ответа.



Закончив работу с *Тренировкой*, результат продемонстрируйте преподавателю.

### 1.2. Выполнение заданий по теме «Система счисления – Целые числа», Зачет:

- Выберите тип занятия «Зачет» – «ОК»  
Закончив работу с *Зачетом*, результат продемонстрируйте преподавателю.

## Задание №2. Работа с Numbers. Вещественные числа.

### 2.2. Выполнение заданий по теме «Система счисления – Вещественные числа». Тренировка.

- Выберите тип занятия «Тренировка» - «ОК»  
Закончив работу с темой, результат продемонстрируйте преподавателю.

**Задание №1.** Работа с Numbers. «Система счисления – Вещественные числа». Зачет.

Результат работы «Зачет» сохраните как копию экрана на флеш-носителе.

**Задание №2.** Переведите следующие десятичные числа в указанную систему счисления, с требуемым количеством знаков после запятой.

Вариант	Число	Система счисления с основанием	Количество знаков после запятой	Система счисления с основанием	Количество знаков после запятой	Число	Система счисления с основанием
1	53,13	8	6	11	4	2,(3)	2
2	40,31	16	5	7	6	1,(3)	2
3	81,11	8	7	12	5	0,(3)	8
4	50,69	16	5	6	7	1,(6)	2
5	41,23	8	8	13	5	0,(9)	8
6	50,27	16	6	7	6	1,(8)	2
7	63,16	8	7	14	4	1,(7)	2
8	42,81	16	5	6	7	0,(7)	8
9	71,03	8	8	11	5	1,(4)	2
10	48,77	16	6	7	6	2,(1)	2
11	61,51	8	6	12	4	0,(4)	8
12	70,13	16	5	6	7	1,(1)	2
13	49,97	8	7	13	4	2,(2)	2
14	81,09	16	6	7	6	0,(8)	8

**Задание №3.** Переведите следующие числа в десятичную систему счисления:

Вариант	1	2	3	4	5	6	7
Число	0,(1001) <sub>2</sub>	0,(0011) <sub>2</sub>	4,(21) <sub>8</sub>	1,(100) <sub>2</sub>	0,(0011) <sub>2</sub>	3,(53) <sub>8</sub>	1,(B) <sub>16</sub>
Вариант	8	9	10	11	12	13	14
Число	2,(A) <sub>16</sub>	0,(1101) <sub>2</sub>	5,(11) <sub>8</sub>	3,(7) <sub>16</sub>	1,(011) <sub>2</sub>	7,(33) <sub>8</sub>	A,(C) <sub>16</sub>

**Задания №2 и №3 оформите как отчет в следующем виде:**

Самостоятельная работа студента по теме  
*Традиционные системы счисления.*  
*Перевод чисел из одной системы счисления в другую.*  
ВАРИАНТ №\_\_

**Задание №2.** Переведите следующие десятичные числа в указанную систему счисления, с требуемым количеством знаков после запятой.

*Решение:*

.....

*Ответ:* .....

**Задание №3.** Переведите следующие числа в десятичную систему счисления.

*Решение:*

.....

*Ответ:* .....

**Смешанные системы счисления**

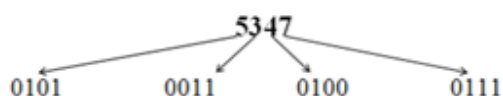
Система счисления (позиционная СС), в которой каждая цифра числа, заданного в Q-ичной СС, заменяется соответствующим представлением в P-ичной СС, называется смешанной или P-Q-ичной СС, (P<Q).

Одним из примеров смешанной системы счисления является двоично-десятичная СС.

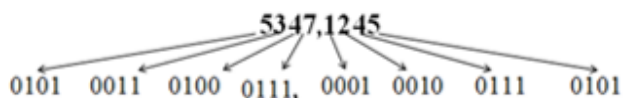
**Двоично-десятичная система счисления**

Каждая цифра десятичного числа записывается (кодируется) четырьмя двоичными цифрами.

**Пример 1.**



**Пример 2.**



Двоично-десятичное представление – это двоичный код для представления десятичного числа, но не равное ему значение в двоичной системе счисления.

Для любого числа в системе счисления с основанием  $P = 2^n$ , смешанный двоично-P-ичный код совпадает с представлением этого числа в двоичной системе счисления. К таким системам счисления относятся двоично-восьмеричная, двоично-шестнадцатеричная и др..

Таблица 1

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



### Двоично-восьмеричная система счисления

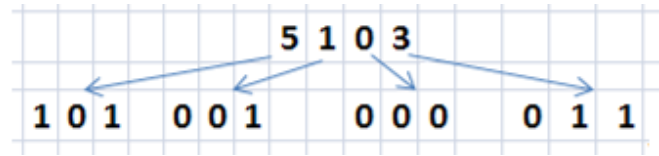
**Пример 1.** Перевод из двоичной системы счисления в восьмеричную; целые числа  
 $10111101_2 = ?_8$

Т.к.  $2^3 = 8$ , то для перевода используются "триады" цифр двоичного числа:

$$\begin{array}{ccc} \underline{010} & \underline{111} & \underline{101} \\ 2 & 7 & 5_8 \end{array}$$

$$10111101_2 = 275_8$$

**Пример 3.** Перевод из восьмеричной системы счисления в двоичную; целые числа  
 $5103_8 = ?_2$



$$5103_8 = 101001000011_2$$

**Пример 2.** Перевод из двоичной системы счисления в восьмеричную; вещественные числа

$$10010110,0111_2 = ?_8$$

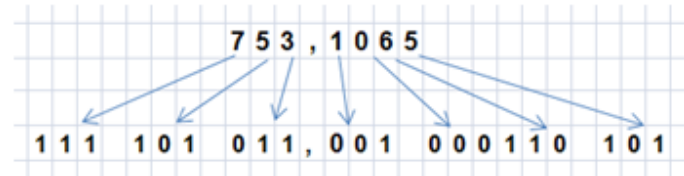
$$10010110,0111_2 = \underline{010} \underline{010} \underline{110}, \underline{011} \underline{100}_2$$

=

$$= 2 \ 2 \ 6 , 3 \ 4_8$$

$$10010110,0111_2 = 226,34_8$$

**Пример 4.** Перевод из восьмеричной системы счисления в двоичную; вещественные числа  
 $753,1065_8 = ?_2$



$$753,1065_8 = 111 \ 101 \ 011, 001 \ 000 \ 110 \ 101_2$$

### Двоично-шестнадцатеричная система счисления

**Пример 5.** Перевод из двоичной системы счисления в шестнадцатеричную; целые числа

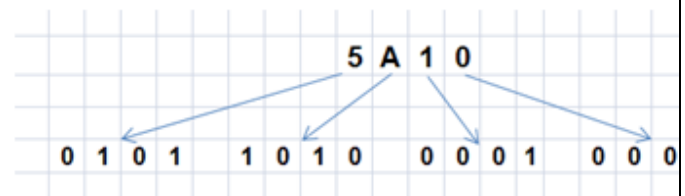
$$110111101_2 = ?_{16}$$

Т.к.  $2^4 = 16$ , то для перевода используются "тетрады" цифр двоичного числа:

$$\begin{array}{ccc} \underline{0001} & \underline{1011} & \underline{1101} \\ 1 & B & D_{16} \end{array}$$

$$110111101_2 = 1BD_{16}$$

**Пример 7.** Перевод из шестнадцатеричной системы счисления в двоичную; целые числа  
 $5A10_{16} = ?_2$



$$5A10_{16} = 0101 \ 1010 \ 0001 \ 0000_2$$

**Пример 6.** Перевод из двоичной системы счисления в восьмеричную; вещественные числа

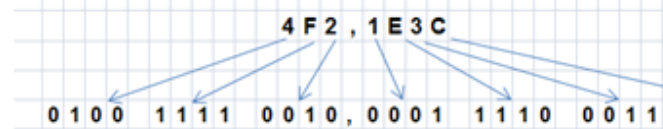
$$1010110,01011_2 = ?_{16}$$

$$\frac{0101}{5} \frac{0110}{6}, \frac{0101}{5} \frac{1000}{8}_{16}$$

$$1010110,01011_2 = 56,58_{16}$$

**Пример 8.** Перевод из шестнадцатеричной системы счисления в двоичную; вещественные числа

$$4F2,1E3C_{16} = ?_2$$



$$4F2,1E3C_{16} = 0100\ 1111\ 0010,0001\ 1110\ 0011\ 1100_2$$

**Пример 9.** Перевод из троичной системы счисления в девятеричную; целые числа.

$$12012_3 = ?_9$$

$$Q = P^n, Q = 9; P = 3, n = 2; 9 = 3^2$$

По таблице Приложения 1 данного раздела получается следующий результат:

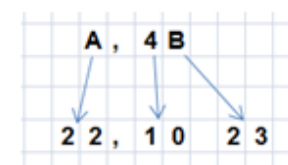
$$\frac{01}{1} \frac{20}{6} \frac{12}{5}_9$$

$$12012_3 = 165_9$$

**Пример 10.** Перевод из шестнадцатеричной системы счисления в четверичную; вещественные числа

$$A,4B_{16} = ?_4$$

По таблице Приложения 2 данного раздела получается следующий результат:



$$A,4B_{16} = 22,1023_4$$

## Нетрадиционные системы счисления

*Традиционная система счисления* – это система счисления, в которой базис состоит из членов геометрической прогрессии.

$$\dots, P^{-2}, P^{-1}, P^0, P^1, P^2, P^3, \dots, P^n \dots$$

Знаменатель **P** геометрической прогрессии называется *основанием* системы счисления.

**Нетрадиционные системы счисления** – это системы счисления, в которых *базис не является геометрической прогрессией*.

Нетрадиционные системы счисления:

1. **Факториальная**
2. **Фибоначчиева**
3. **Троичная уравновешенная**

### 1. Факториальная система счисления

**Алфавит:** зависит от числа используемых разрядов: если используется  $n$  разрядов, то количество цифр равно  $n+1$  (включая ноль). Размерность алфавита увеличивается с ростом номера разряда. В первом разряде могут быть только цифры 0 и 1, во втором: 0,1, 2, в  $k$ -м – 0,1,2,  $k$  и т.д.

**Базис:** последовательность факториалов: натуральных чисел:  $1!, 2!, 3!, \dots, n!, \dots$

**Основание:** последовательность факториалов.

Если запись числа в факториальной системе имеет вид:  $d_n d_{n-1} \dots d_2 d_1$ , то этому числу соответствует десятичное значение следующего вида:

$$\sum_{k=1}^n d_k k! = d_1 * 1! + d_2 * 2! + d_3 * 3! + \dots + d_n * n!$$

, где  $d_k$  цифра числа,  $0 \leq d_k \leq k$

**Пример 11.** Перевод из десятичной системы счисления в факториальную

$$89_{10} = ?_f$$

$$\begin{array}{r|l} 89 & 2 \\ \hline 88 & 44 \\ \hline 1 & 42 \\ & 2 \\ & 14 \\ & 3 \\ & 12 \\ & 2 \\ & 3 \end{array}$$

$$89_{10} = 3221_f$$

**Пример 12.** Перевод из факториальной системы счисления в десятичную

$$4021_f = ?_{10}$$

$$4! 3! 2! 1! \\ 4 \ 0 \ 2 \ 1_f = 4 * 4! + 0 * 3! + 2 * 2! + 1 * 1! = 101_{10}$$

$$4021_f = 101_{10}$$

### 2. Фибоначчиева система счисления

**Алфавит:** цифры 0 и 1.

**Базис:** последовательность чисел Фибоначчи: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 ...

**Основание:** нет.

Получение числа в Фибоначчиевой системе счисления – это последовательное выделение из заданного числа слагаемых, равных наибольшему возможным числам Фибоначчи.

**Пример 13.** Перевод из десятичной системы счисления в фибоначчьеву

$$25_{10} = ?_{fib}$$

1. Ряд Фибоначчи:

1 2 3 5 8 13 21

2. Выделение слагаемых:  $25 = 21 + 3 + 1$

3. Записывается "1", если число из ряда есть в выделенной сумме, "0" - если нет:

1 2 3 5 8 13 21

1 0 1 0 0 0 1

$$25_{10} = 1000101_f$$

**Пример 12.** Перевод из фибоначчьевой системы счисления в десятичную

$$10100_{fib} = ?_{10}$$

8 5 3 2 1

$$1 \ 0 \ 1 \ 0 \ 0_{fib} = 1 * 8 + 0 * 5 + 1 * 3 + 0 * 2 + 0 * 1 = 11_{10}$$

$$10100_{fib} = 11_{10}$$

### 3. Троичная уравновешенная система счисления

Алфавит: цифры 0, 1, -1.

Основание: 3.

Особенность уравновешенных систем счисления - перед отрицательными числами не ставится знак "минус" и при выполнении арифметических операций не используется "правило знаков".

Отрицательные числа		Положительные числа	
10-ичная система счисления	Троичная уравновешенная система счисления	10-ичная система счисления	Троичная уравновешенная система счисления
-1	-1	1	1
-2	-1 1	1 -1	2
-3	-1 0	1 0	3
-4	-1 -1	1 1	4
-5	-1 1 1	1 -1 -1	5

**Пример 14.** Перевод из десятичной системы счисления в троичную уравновешенную

$$-25_{10} = ?_{\text{ур}}$$

$$\begin{array}{r|l}
 -25 & \frac{3}{-8} \\
 \hline
 -24 & \frac{3}{-3} \\
 \hline
 -1 & \frac{3}{-1} \\
 \hline
 & 1 \\
 & \frac{3}{-3} \\
 & \hline
 & 0 \\
 & \hline
 & -1
 \end{array}$$

$$-25_{10} = -101-1_{\text{ур}}$$

**Пример 15.** Перевод из троичной уравновешенной системы счисления в десятичную

$$-10-100_{\text{ур}} = ?_{10}$$

$$\begin{array}{r}
 4 \ 3 \ 2 \ 1 \ 0 \\
 -1 \ 0 \ -1 \ 0 \ 0_{\text{ур}} = (-1) \cdot 3^4 + 0 \cdot 3^3 + \\
 + (-1) \cdot 3^2 + 0 \cdot 3^1 + 0 \cdot 3^0 = -90_{10}
 \end{array}$$

$$-10-100_{\text{ур}} = -90_{10}$$

#### Задания для выполнения в аудитории

**Задание №1.** Выполните преобразования следующих чисел:

1.1. Из факториальной в 10-ичную: **4310**.

1.2. Из 10-ой системы счисления в фибоначчеву: **1286**.

1.3. Из 10-ой системы счисления в уравновешенную троичную: **155**.

**Задание №1. Выполните преобразования следующих чисел:**

Вариант	Из факториальной системы счисления в 10-ичную	Из 10-ичной системы счисления в факториальную	Из Фибоначчиевой системы счисления в 10-ичную	Из 10-ичной системы счисления в фибоначчиеву	Из уравновешенной троичной системы счисления в десятичную	Из 10-ичной системы счисления в уравновешенную троичную	Из 10-ичной системы счисления в двоично-десятичную
1	12201	64	10101	88	-1-101	-45	1101101,014
2	12310	65	100100	103	-11-1-1	-48	1101111,023
3	43110	74	10000	92	1-10-1	-20	1001101,201
4	11120	45	1000100	87	1-1-1-10	-49	1001100,503
5	31110	58	101001	102	-10-11	-21	1111101,071
6	21100	61	100100	89	1-10-10	-46	1010101,809
7	11220	71	101010	105	1-100	-22	1110001,612
8	13210	53	101000	91	-11-10	-19	1010001,872
9	53210	62	1010001	86	110-1-1	-47	10100001,111
10	13210	78	1000000	93	-11-1-10	-23	11100010,242
11	41210	42	100000	100	-10-10	-53	1000001,882
12	23210	67	100001	95	-10-1-1	-24	1000021,747
13	43200	54	100101	77	-1101	-52	1000301,373
14	53211	72	10010	97	-1100	-42	1004001,212
15	42210	69	100001	82	-1-100	-51	1400001,454

**Задания оформите как отчет в следующем виде:**

Самостоятельная работа студента по теме  
*Смешанные и Нетрадиционные системы счисления.*  
*Перевод чисел из одной системы счисления в другую.*

ВАРИАНТ №\_\_

**Задание №1.** Выполните преобразования следующих чисел:

**Из факториальной системы счисления в десятичную систему счисления.**

$$31210_f = X_{10}$$

**Решение:**

.....

**Ответ:** .....

---

#### ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

---

1. Андреева Е., Фалина И. Информатика: Системы счисления и компьютерная арифметика. М.: Лаборатория базовых знаний, 1999. 256 с.
2. Национальный открытый университет. Лекция 4: "Системы счисления и действия в них".  
URL: <https://www.intuit.ru/studies/courses/108/108/lecture/3143>

## Приложение 1

---

Таблица соответствия чисел в десятичной, троичной и девятеричной системах счисления

«10»	«3»	«9»
0	<b>0</b>	<b>0</b>
1	<b>1</b>	<b>1</b>
2	<b>2</b>	<b>2</b>
3	10	<b>3</b>
4	11	<b>4</b>
5	12	<b>5</b>
6	20	<b>6</b>
7	21	<b>7</b>
8	22	<b>8</b>
9	100	10

## Приложение 2

---

Таблица соответствия чисел в десятичной, четверичной и шестнадцатеричной системах счисления

«10»	«4»	«16»
0	0	0
1	1	1
2	2	2
3	3	3
4	10	4
5	11	5
6	12	6
7	13	7
8	20	8
9	21	9
10	22	A
11	23	B
12	30	C
13	31	D
14	32	E
15	33	F



## МАШИННОЕ ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

Для хранения чисел в памяти компьютера используется:

1. Целочисленный формат (естественная форма):
  - целых положительных чисел (без знака);
  - целых положительных чисел (со знаком).
2. Формат с плавающей точкой (нормализованная форма) для хранения вещественных чисел.

### ПРЕДСТАВЛЕНИЕ ЦЕЛЫХ ЧИСЕЛ В ПАМЯТИ КОМПЬЮТЕРА

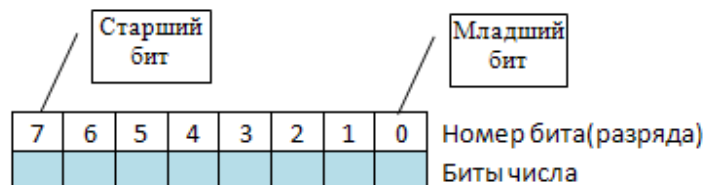
#### Форматы представления целых чисел

Формат – это соглашение или правила представления числа в виде последовательности бит.

Форматы представления целых чисел:

- полуслово ( 1 байт);
- слово (2 байта);
- двойное слово (4 байта);
- расширенное слово (8 байт).

#### Машинное представление целых положительных чисел (без знака)



**Пример 1.** Запишите внутреннее представление числа 18 в однобайтовом и двухбайтовом форматах беззнаковых целых чисел.

**Решение:**

*Однобайтовый формат:*

1. Перевести число в двоичную систему счисления:  $18_{10} = 10010_2$
2. Заполнить значения бит числа, начиная с младшего бита, дополнив оставшиеся биты нулями:

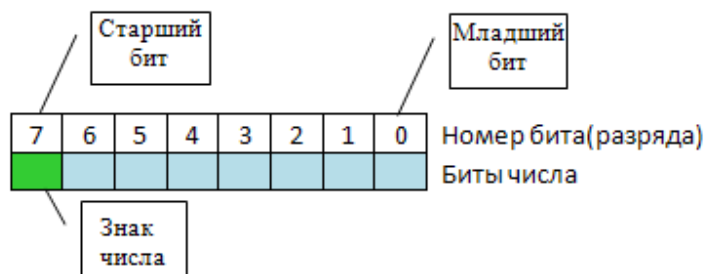
7	6	5	4	3	2	1	0
0	0	0	1	0	0	1	0

*Двухбайтовый формат:*

1. Перевести число в двоичную систему счисления:  $18_{10} = 10010_2$
2. Заполнить значения бит числа, начиная с младшего бита, дополнив оставшиеся биты нулями:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
Старший байт								Младший байт							

### Машинное представление целых отрицательных чисел (со знаком)



Старший бит (разряд) содержит информацию о знаке числа. Если число положительное, то в старшем бите хранится ноль, если число отрицательное, то в старшем бите хранится единица. Для двоичного представления самого числа, например для однобайтового формата, отводятся биты с нулевого по шестой.

**Пример 2.** Запишите внутреннее представление числа **-18** в однобайтовом и двухбайтовом форматах знаковых целых чисел.

**Решение:**

*Однобайтовый формат:*

1. Число «-18» < 0, следовательно значение старшего бита (7-го бита) = «1»

7	6	5	4	3	2	1	0
1							

2. Получить **Прямой код** числа:

2.1. Перевести  $|-18|$  в двоичную систему счисления:  $18_{10} = 10010_2$

2.2. Заполнить значения бит числа, начиная с младшего бита, дополнив оставшиеся биты нулями:

7	6	5	4	3	2	1	0
1	0	0	1	0	0	1	0

3. Получить **Обратный код** числа – инвертирование разрядов («0» => «1»; «1» => «0»), кроме разряда со знаком:

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1

4. Получить **Дополнительный код** числа – к обратному коду прибавляется единица 1101101 +1:

7	6	5	4	3	2	1	0
1	1	1	0	1	1	0	1
+ 1							
1	1	1	0	1	1	1	0

*Двухбайтовый формат:*

1. Значение старшего бита (15-го бита) = «1»:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1															

2. Получить **Прямой код** числа:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

3. Получить **Обратный код** числа:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1

4. Получить **Дополнительный код** числа: «Обратный код» +1:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0

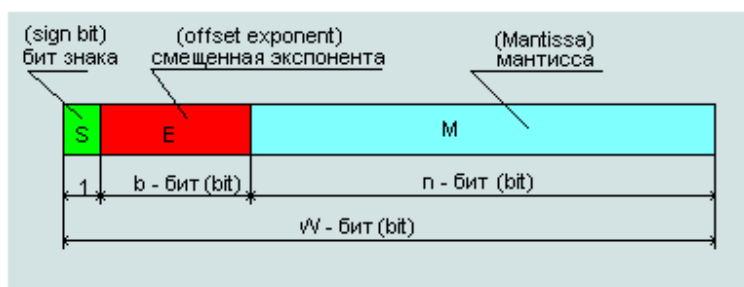
## ПРЕДСТАВЛЕНИЕ ВЕЩЕСТВЕННЫХ ЧИСЕЛ В ПАМЯТИ КОМПЬЮТЕРА

Стандарт IEEE 754 (стандарт двоичной арифметики с плавающей точкой) регламентирует машинное представление чисел с плавающей точкой, а также базовые операции над ними: сложение, вычитание, умножение, деление, сравнение, приведение типов. Из математических функций этим стандартом регламентируется только извлечение квадратного корня. IEEE 854 обобщает стандарт IEEE 754 и регламентирует представление десятичных чисел с плавающей точкой. Позднее требования этих стандартов были отражены в стандарте ИЕС 60559:1989 (двоичная арифметика с плавающей точкой для микропроцессорных систем). В 2008 г. ассоциация IEEE выпустила стандарт IEEE 754-2008, который включил в себя стандарт IEEE 754-1985.

Стандарты IEEE 754 и IEEE 854 определяют несколько возможных форматов чисел с плавающей точкой:

1. *одинарной точности* (single precision);
2. *двойной точности* (double precision);
3. *расширенной двойной точности* (double-extended precision).

Формальное представление чисел в стандарте IEEE 754 для любого формата точности представлено ниже:



На все число, с учетом знака отводится **W** бит.

**S** – бит знака, если  $S=0$  – положительное число;  $S=1$  – отрицательное число

**E** – смещенная экспонента двоичного числа (**b** – количество бит для записи экспоненты);

$\text{exp}_2 = E - (2^{(b-1)} - 1)$  – экспонента двоичного нормализованного числа с плавающей точкой

$(2^{(b-1)} - 1)$  – заданное смещение экспоненты

**M** – остаток мантиссы двоичного нормализованного числа с плавающей точкой (**n** – количество бит для записи остатка мантиссы)

Для чисел:

1. *однократной точности* (single precision):  $W = 32$ ;  $b = 8$ ;  $n = 23$ ; смещение экспоненты равно  $(2^{(b-1)} - 1) = (2^{(8-1)} - 1) = 127$ .
2. *двойной точности* (double precision):  $W = 64$ ;  $b = 11$ ;  $n = 52$ ; смещение экспоненты равно 1023.
3. *расширенной двойной точности* (double-extended precision): значения  $W$  и  $b$  в стандартах не фиксируются, вводятся ограничения:  
 $128 \geq W \geq 80$  и  $b \geq 15$ . (В процессорах 32-битной архитектуры Intel  $W = 80$ ,  $b = 15$ ,  $n=64$ ; смещение экспоненты равно 16383).

Иногда используются числа *четырёхкратной точности*:  $W = 128$ ;  $b = 15$ ;  $n = 112$ ; смещение экспоненты равно 16383.

**Пример 1.** Запишите внутреннее представление числа - **506,625** в формате однократной точности.

**Решение:**

1. Формат однократной точности, следовательно:  $W = 32$ ;  $b = 8$ ;  $n = 23$ ; смещение экспоненты равно 127.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	...	4	3	2	1	0
1																		...					
Смещенная экспонента									Мантисса														

2. Число отрицательное, следовательно, бит  $31 = 1$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	...	8	7	6	5	4	3	2	1	0
1														...									

3. Модуль числа переводится в систему счисления с основанием 2:

$$506,625D = 11111010,101B$$

4. Число приводится к нормализованному виду:

$$11111010,101B = 1,11111010101B * 2^{+8}$$

5. Рассчитывается смещенный порядок:

$E = \text{истинный порядок} + \text{смещение экспоненты};$

$$E = 8 + 127 = 135D = 10000111B$$

Записывается смещенный порядок в разряды с 30 по 23.

6. Записывается остаток мантиссы числа **1,11111010101** в разряды с 22 по 0, заполняя пустые разряды нулями:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	...	2	1	0
1	1	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	...	0	0	0

**Пример 2.** Запишите внутреннее представление числа - **506,625** в формате двойной точности.

**Решение:**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	...	...	2	1	0
1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	0	...	...	0	0	0

**Пример 3.** Дано внутреннее представление числа - 506,625 в формате однократной точности (Пример 1). Осуществить проверку получения десятичного числа - 506,625 из чисел, представленного в стандарте IEEE754.

**Решение:**

Вычисление десятичных чисел с плавающей точкой, из чисел, представленных в стандарте IEEE754, осуществляется по следующей формуле:

$$F = (-1)^S 2^{(E-2^{(b-1)+1})} (1+M/2^n)$$

S – бит знака; E – смещенная экспонента двоичного числа; b – количество бит для записи экспоненты;

M – остаток мантииссы двоичного нормализованного числа с плавающей точкой; n – количество бит для записи остатка мантииссы.

Для числа "-506, 625" из Примера 1:

S=1(разряд 31); E = 10000111B (разряды с 30 по 23) => 135D;

M = 111110101010000000000000B(разряды с 22 по 0) => 8212480D; n = 23.

F = 506,62500000000000000000 D

**Задание №1.**

Запишите внутреннее представление числа **141D** в двухбайтовом формате беззнаковых целых чисел.

**Задание №2.**

Запишите внутреннее представление чисел **25D** и **-9D** в однобайтовом и двухбайтовом форматах знаковых целых чисел.

**Задание №3. (2 балла)**

Запишите внутреннее представление десятичных чисел в формате однократной точности:

<i>Вариант</i>	<i>Числа</i>	
<b>1</b>	<b>0,3</b>	<b>- 11,3</b>
<b>2</b>	<b>0,5</b>	<b>- 12,5</b>
<b>3</b>	<b>1,1</b>	<b>- 14,1</b>
<b>4</b>	<b>0,09</b>	<b>- 13,09</b>
<b>5</b>	<b>1,2</b>	<b>- 15,2</b>
<b>6</b>	<b>0,7</b>	<b>- 17,7</b>
<b>7</b>	<b>1,04</b>	<b>- 21,04</b>
<b>8</b>	<b>0,2</b>	<b>- 20,2</b>
<b>9</b>	<b>1,3</b>	<b>- 18,3</b>
<b>10</b>	<b>0,4</b>	<b>- 24,4</b>
<b>11</b>	<b>1,5</b>	<b>- 15,5</b>
<b>12</b>	<b>0,08</b>	<b>- 28,08</b>
<b>13</b>	<b>1,7</b>	<b>- 14,7</b>
<b>14</b>	<b>1,8</b>	<b>- 10,8</b>

**Задание №1. "Тренинг"(1 балл).**

Запишите внутреннее представление чисел в форматах однократной и двойной точности. Проверьте правильность полученного решения на сайте <http://www.binaryconvert.com> – Online Binary-Decimal Converter (Двоично-десятичный конвертор согласно стандарту IEEE754).

<b>Вариант</b>	<b>Числа</b>	
<b>1</b>	<b>30,01</b>	<b>– 421,23</b>
<b>2</b>	<b>40,05</b>	<b>– 132,85</b>
<b>3</b>	<b>41,14</b>	<b>– 214,21</b>
<b>4</b>	<b>31,19</b>	<b>– 313,69</b>
<b>5</b>	<b>51,25</b>	<b>– 415,52</b>
<b>6</b>	<b>60,17</b>	<b>– 517,57</b>
<b>7</b>	<b>71,84</b>	<b>– 231,77</b>
<b>8</b>	<b>90,82</b>	<b>– 520,52</b>
<b>9</b>	<b>81,73</b>	<b>– 618,43</b>
<b>10</b>	<b>78,34</b>	<b>– 224,94</b>
<b>11</b>	<b>91,75</b>	<b>– 815,75</b>
<b>12</b>	<b>70,98</b>	<b>– 128,99</b>
<b>13</b>	<b>61,67</b>	<b>– 414,27</b>
<b>14</b>	<b>99,18</b>	<b>– 710,78</b>



**Задание №2. (3 балла)**

Дано внутреннее представление числа в стандарте IEEE754.

Определите: 1) формат представления числа; 2) какое десятичное число записано в памяти компьютера.

Вариант	Числа																											
1	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	...	2	1	0		
	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	...	0	0	0		
2	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	...	2	1	0		
	0	1	0	1	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	1	0	...	0	0	0		
3	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	...	1	0		
	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	1	1	1	0	...	0	0		
4	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	...	3	2	1	0		
	0	1	0	0	1	1	0	0	1	0	0	0	1	0	1	1	0	0	0	1	0	...	0	0	0	0		
5	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	...	1	0	
	1	1	1	0	1	0	1	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1	1	0	...	0	0
6	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	...	1	0	
	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	0	0	1	0	1	0	0	1	0	...	0	0	
7	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	...	1	0		
	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	1	0	0	...	0	0		
8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	...	0	
	1	1	1	1	0	0	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	...	0
9	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	...	2	1	0		
	1	1	0	1	1	0	0	1	0	0	0	0	1	0	1	0	1	0	1	0	0	0	...	0	0	0		
10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	...	1	0		
	0	1	0	1	0	1	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	...	0	0		
11	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	...	1	0		
	1	1	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0	...	0	0		
12	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	...	1	0		
	0	1	0	0	0	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	0	1	0	...	0	0		
13	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	...	1	0		
	1	1	1	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	0	0	0	1	0	...	0	0		
14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	...	1	0		
	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	...	0	0		

## ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

---

1. Национальный Открытый Университет ИНТУИТ. Двоичное представление целых чисел. URL: <https://www.intuit.ru/studies/courses/569/425/lecture/9671?page=1#sect2>.
2. Поляков К.Ю., Шестаков А.П., Еремин Е.А. Компьютерная арифметика // "Информатика" №11, 2011. URL: <http://информатика.1сентября.рф/index.php?year=2011&num=01>.
3. Интерактивный двоично-десятичный конвертер. URL: <http://www.binaryconvert.com/index.html>.
4. IEEE 754-1985. IEEE Standard for Binary Floating-Point Arithmetic. N.Y.: IEEE, 1985.
5. IEEE 854-1987. IEEE Standard for Radix-Independent Floating-Point Arithmetic. N.Y.: IEEE, 1987.
6. Каталог МЭК — публикации Международной электротехнической комиссии. ИЕС./ИЕС 60559(1989). Двоичная арифметика с плавающей запятой для микропроцессорных систем. URL: <http://en.gostinfo.ru/catalog/ieclist>.
7. Online Binary-Decimal Converter (Двоично-десятичный конвертер, согласно стандарту IEEE754). URL: <http://www.binaryconvert.com>.
8. Проект "Личности". URL: [https://persons-info.com/persons/GORNER\\_Uiliam\\_Dzhordzh/7717](https://persons-info.com/persons/GORNER_Uiliam_Dzhordzh/7717).

## ИЗМЕРЕНИЕ ИНФОРМАЦИИ

---

### *Единицы измерения информации*

1 бит

1 байт = 8 бит

1 Килобайт (Кб) = 1024 байта =  $2^{10}$  байт

1 Мегабайт (Мб) = 1024 Кб =  $2^{10}$  Кбайт

1 Гигабайт (Гб) = 1024 Мб =  $2^{10}$  Мбайт

1 Терабайт (Тб) = 1024 Гб =  $2^{10}$  Гбайт

1 Петабайт (Пб) = 1024 Тб =  $2^{10}$  Тбайт

1 Эксабайт (Эб) = 1024 Пб =  $2^{10}$  Пбайт

1 Зеттабайт (Зб) = 1024 Эб =  $2^{10}$  Эбайт

1 Йоттабайт (Йб) = 1024 Зб =  $2^{10}$  Збайт

Например:

- 100 Кбайт занимает снимок в телефоне с низким разрешением.
- 1 Мбайт – небольшая художественная книга.
- 3 гигабайта – 1 час видеозаписи в хорошем качестве.
- 1 гигабайт текста способен прочитать человек за всю свою жизнь.

Измерение информации рассматривают:

для равновероятных событий: *содержательный подход; алфавитный подход;*

для неравновероятных событий : *содержательный подход; алфавитный подход.*

### СОБЫТИЯ РАВНОВЕРОЯТНЫЕ

---

*События равновероятны, если ни одно из них не имеет преимущества перед другими.*

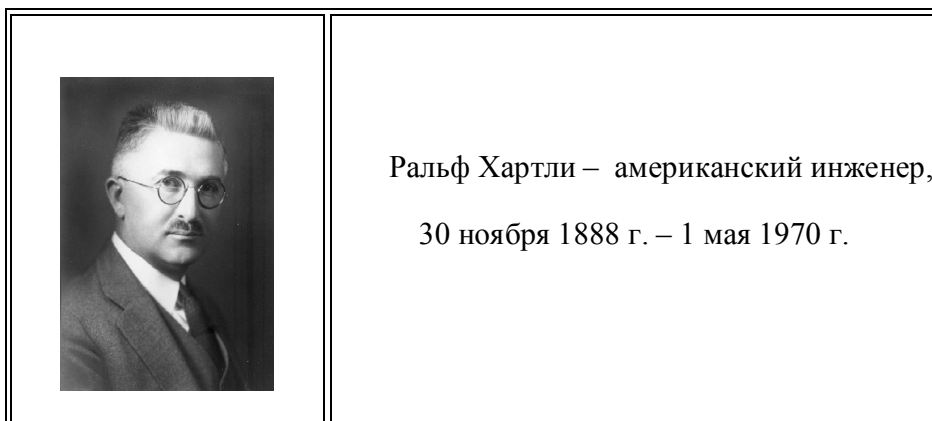
С этой точки зрения выпадения орла и решки — равновероятны; выпадения каждой из шести граней кубика — равновероятны.

#### **1. Содержательный подход**

Если известно, что может произойти одно из  $N$  равновероятных событий, то свершившийся факт (событие произошло), несет количество информации, определяемое по следующей формуле (*формула Хартли*):

$$i = \text{Log}_2 N \quad (N=2^i),$$

где  $N$  – количество равновероятных событий;  $i$  – количество информации в одном таком событии.



**Количество информации в сообщении:**

$$I = K_{\text{собр}} * i,$$

где  $I$  – Количество информации в сообщении;  $K_{\text{собр}}$  – количество равновероятных событий;

$i$  – количество информации в одном таком событии

**Пример 1.** Какое количество информации содержит сообщение о том, что в лотерее выпали шары с номерами 5, 18, 40, 51, если всего разыгрывалось 64 шара с номерами от 1 до 64?

*Дано:*  $N=64$ .

*Найти:*  $I$  – ?

-----  
**Решение:**

Количество информации в сообщении:  $I = K_{\text{собр}} * i$

$$i = \log_2 N, \quad N = 2^i$$

$$K_{\text{собр}} = 4; \quad (5, 18, 40, 51)$$

$i = \log_2 64$ , следовательно,  $i = 6$  бит, тогда:  $I = K_{\text{собр}} * i = 4 * 6 = 24$  бит.  **$I = 24$  бита**

**Пример 2.** Сколько информации несет сообщение о том, что было угадано число в диапазоне целых чисел от 67 до 1090?

Дано: диапазон чисел от 67 до 1090.

Найти:  $i$  –?

-----  
**Решение:**

$$N=2^i$$

$$N= 1090-67+1=1024$$

$i = \text{Log}_2 1024$ , следовательно,  **$i=10$  бит**

**Пример 3.** В лотерее выпало 6 шаров из  $N$  возможных и сообщение об этом несет 42 бит информации. Чему равно  $N$  ?

Дано:  $K_{\text{собр}}=6$ ;  $I=42$ .

Найти:  $N$  – ?

-----  
**Решение:**

$$N=2^i$$

$i$ - вычисляется из формулы:  $I=K_{\text{собр}} * i$  :

$i = I / K_{\text{собр}}$ , следовательно,  $i = 42/6 = 7$  бит информации о том, что выпал 1 шар

$$N = 2^i \Rightarrow 2^7 = 128; \quad N = 128$$

## 2. Алфавитный подход

Алфавитный подход определяет количество информации в тексте (объективный подход).

Количество информации рассчитывается с использованием формулы:

$$N=2^i$$

В качестве  $N$  – размерность алфавита (мощность алфавита  $M$ );  $i$ - кол-во информации в одном символе алфавита.

Количество информации в текстовом сообщении:

$$I=K_{\text{симв}} * i, \text{ где}$$

$I$  – Количество информации в сообщении;  $K_{\text{симв}}$  – количество символов в сообщении;

$i$  – количество информации в одном символе алфавита, с помощью которого записано сообщение;

$$I=K_{\text{симв}} * i = K_{\text{симв}} * \text{Log}_2 M$$

Максимальное количество слов  $L$ , которое можно составить из  $m$  букв, с помощью алфавита мощностью  $N$ , определяется по формуле:

$$L=N^m$$

**Пример 4.** Информационное сообщение объемом 1,5 Кб содержит 3072 символа. Сколько символов содержит алфавит, при помощи которого было записано это сообщение?

*Дано:*

$I = 1,5 \text{ Кб}; K = 3072 \text{ символа}$

*Найти:* мощность алфавита

$N - ?$

-----

**Решение:**

Мощность алфавита определяется по формуле:  $N = 2^i$ ;

$I = K \cdot i$ , следовательно количество информации в одном символе алфавита  $i = I / K$ ;

$I = 1,5 \cdot 1024(\text{Байт}) \cdot 8 = 12288 \text{ бит}$ , тогда  $i = 12288 / 3072 = 4 \text{ бита}$

$N = 2^4 = 16 \text{ символов}$ .  **$N = 16 \text{ символов}$ .**

**Пример 5.** Мощность алфавита равна 256. Сколько Кбайт памяти потребуется для сохранения 160 страниц текста, содержащего в среднем 192 символа на каждой странице?

*Дано:*

Мощность алфавита  $N = 256$

Количество страниц: 160

Количество символов на странице: 192

*Найти:*

$I - ?$

-----

**Решение:**

Количество информации в сообщении будет равно объему памяти, необходимому для хранения 160 страниц текста.

$I = K_{\text{СИМВ}} \cdot i$

$K_{\text{СИМВ}} = \text{Кол страниц} \cdot \text{Кол симв на стр} = 160 \cdot 192 = 30\,720 \text{ символов}$ ;

$i$  определяется из формулы:  $N = 2^i$ ;

$256 = 2^i$ , следовательно  $i = 8 \text{ бит}$ ; тогда количество информации вычисляется следующим образом:

$I = K_{\text{СИМВ}} \cdot i = 30\,720 \cdot 8 = 245\,760 \text{ бит}$ .

Перевод в Байты:  $(30\,720 \cdot 8) / 8$ . Перевод в КБайты:  $((30\,720 \cdot 8) / 8) / 1024 = 30 \text{ КБайт}$ .

Необходимый объем памяти равен **30 КБайт**.

## Задания для выполнения в аудитории

---

### **Задание №1**

Объем сообщения равен 11 Кбайт. Сообщение содержит 11264 символа. Какова мощность алфавита, с помощью которого было записано сообщение?

### **Задание №2**

Растровое графическое изображение 30x30 точек содержит не более 256 цветов. Сколько Байт памяти потребуется для хранения изображения?

### **Задание №3**

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

**Задание №1. Тренировка в программе "Измерение информации" – равновероятные события.**

- Запустите тренажер "Измерение информации".
- Выберите тему "Измерение информации".
- Результат сохраните как копию экрана на флеш-носителе.

**Задание №2**

**1. Определите, какой подход используется для измерения информации в предложенных задачах?**

**2. Решите задачу соответственно варианту. Оформите решение с подробными вычислениями и пояснениями.**

№ Варианта	Задача
1	ДНК человека (генетический код) можно представить как некоторое слово в четырехбуквенном алфавите. Каждой букве соответствует звено цепи ДНК (нуклеотид). Определите, какое количество информации содержит ДНК, если она состоит примерно из $1.5 \times 10^{23}$ нуклеотидов? Определите суммарный объем информации в человеческом теле, считая его состоящим примерно из $10^{13}$ клеток.
2	Можно ли записать на флеш-носитель (2 Гбайт) книгу, имеющую 323 страницы, причем на каждой странице этой книги 47 строк, а в каждой строке 64 символа?
3	Сообщение из 48 символов было записано в 8-битной кодировке Windows-1251. После вставки в текстовый редактор сообщение было перекодировано в 16-битный код Unicode. На сколько байт увеличится количество памяти для хранения данного сообщения?
4	В велокроссе участвуют 1025 спортсменов. Устройство регистрирует прохождение промежуточного финиша каждым из участников, записывая его номер с использованием минимально возможного количества бит, одинакового для всех номеров. Каков информационный объем сообщения, записанного устройством после того, как промежуточный финиш прошли 200 велосипедистов?
5	Цветное изображение имеет 256 цветов. Размер изображения 9,5*15,5 см. Для хранения изображения требуется 432*105 бит памяти. Каково разрешение изображения в точках на дюйм? (1дюйм=2,5см)
6	Отправлено SMS-сообщение: <i>"Глупое сердце, не бейся! Все мы обмануты счастьем, Нищий лишь просит часть... Глупое сердце, не бейся."</i> В мобильном телефоне адресата установлено ограничение размера входящего SMS-сообщения 64 байтами (при превышении этого размера сообщение автоматически делится на части). Каждый символ кодируется 16 битами. На сколько частей будет разбито сообщение?



№ Вари- анта	Задача
7	Одна ячейка памяти «троичной ЭВМ» может принимать одно из трех возможных состояний. Для хранения некоторой величины отвели 15 ячеек памяти. Сколько значений может принимать эта величина?
8	Растровое графическое изображение 40x40 точек содержит не более 256 цветов. Сколько памяти потребуется для хранения изображения?
9	За 50 секунд был распечатан текст. Подсчитать количество страниц в тексте, если известно, что в среднем на странице 50 строк по 75 символов в каждой. Скорость печати лазерного принтера 1 Кбайт/с.
10	Черно-белое изображение имеет 8 градаций яркости. Размер изображения 15*15 см. Разрешение 300 точек на дюйм (1 дюйм = 2,5 см). Сколько Кбайт памяти требуется для хранения данного изображения?
11	Системный администратор ограничил длительность непрерывного подключения компьютеров сотрудников организации к сети Интернет 20 минутами. Сотруднику требуется переслать файл размером 50 Гбайт. Скорость передачи информации с рабочего места сотрудника в среднем составляет 512 Кбит/с. На сколько частей необходимо разделить файл для пересылки?
12	Имеется 2 текста на разных языках. Первый текст использует 32-символьный алфавит и содержит 280 символов, второй – 16-символьный алфавит и содержит 254 символов. Какой из текстов содержит большее количество информации и на сколько бит?
13	Для регистрации на сайте пользователю необходимо придумать пароль длиной ровно 11 символов. В пароле можно использовать десятичные цифры и 32 различных символа местного алфавита, причем все буквы используются в двух начертаниях – строчные и прописные. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый пароль одинаковым и минимально возможным количеством байт. Каков объем памяти, необходимый для хранения 40 паролей?
14	Лазерный принтер печатает со скоростью в среднем 1 Кбайт в секунду. Сколько времени понадобится для распечатки 26-страничного документа, если известно, что на одной странице в среднем по 49 строк, в строке 63 символа.

## СОБЫТИЯ НЕРАВНОВЕРоятНЫЕ

$N$  – множество элементарных событий

$A$  – одно элементарное событие из  $N$  возможных

$K_A$  - количество элементарных событий из  $N$

$P_A$  – вероятность события  $A$

$$P_A = K_A / N$$

$$0 \leq P_A \leq 1$$

*Вероятность события выражается в долях единицы.*

$I_A$  – количество информации в элементарном событии ( $A$ ):  $I_A = \text{Log}_2 (1/ P_A)$

Чем меньше вероятность  $P$  какого-либо события, тем больше информации в сообщении о данном событии  $I$ .

*Энтропия ( $H$ ) – мера неопределенности, выраженная в битах.*

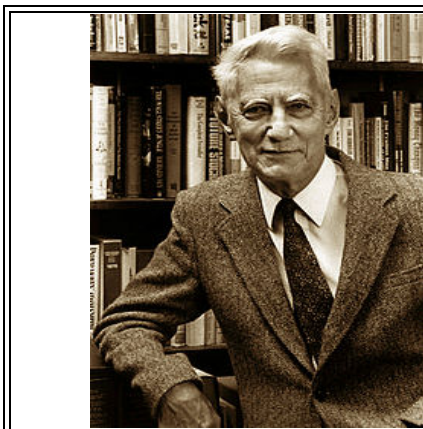
**Энтропíя** в теории информации – мера хаотичности информации, неопределённость свершения какого-либо события, неопределённость появления какого-либо символа первичного алфавита.

**Энтропия** — это количество информации, приходящейся на одно элементарное сообщение источника, вырабатывающего статистически независимые сообщения.

*В общем случае, энтропия  $H$  рассчитывается по формуле Шеннона, предложенной им в 1948 г. в статье «Математическая теория связи»:*

$$H(x) = - \sum_{i=1}^n p(i) \log_2 p(i). \quad I = - \sum_{i=1}^n p_i \log_2 p_i.$$

$N$  – исходное количество рассматриваемых вариантов;  $p(i)$  – вероятность каждого события.



*Клод Шеннон  
американский учёный  
30 апреля 1916 г. – 24 февраля 2001 г.*

Энтропия в формуле Шеннона является средней характеристикой – математическим ожиданием распределения случайной величины  $\{I_0, I_1, \dots, I_{N-1}\}$ .

В случае, когда все варианты *равновероятны*, формула Шеннона совпадает с *формулой Хартли*, которая впервые была предложена американским инженером Ральфом Хартли в 1928г.

### 1. Содержательный подход

**Пример 1.** Студент за семестр получил 32 оценки. Из них: «2» - 4; «3» - 4 ; «4» - 9 ; «5» - 15.

*Определите:*

1. Вероятность получения каждой оценки.
2. Сколько информации в следующих сообщениях:
  - a. «Студент получил 4»
  - b. «Студент получил 2»
  - c. «Студент получил одну из перечисленных выше оценок»

**Решение:**

1. Вероятность получения каждой оценки:

Применяется формула:  $P_A = K_A / N$

Вероятность получения двойки:  $P_2 = (\text{Количество двоек}) / (\text{Общее кол-во оценок})$ ;

$$P_2 = 4/32 = 1/8; \quad P_3 = 4/32 = 1/8; \quad P_4 = 9/32 = 3/14; \quad P_5 = 15/32 = 5/14.$$

2. Сколько информации в следующих сообщениях.

Применяется формула:  $I_A = \text{Log}_2 (1/ P_A)$

a. «Студент получил 4»:  $I_4 = \text{Log}_2 (1/ P_4) = \text{Log}_2 (1/ (3/14)) = \text{Log}_2 (14 / 3) \approx 2.22$  (бит)

b. «Студент получил 2»:  $I_2 = \text{Log}_2 (1/ P_2) = \text{Log}_2 (1/ (1/8)) = \text{Log}_2 (8 / 1) = 3$  (бит)

c. «Студент получил одну из перечисленных выше оценок»:

Применяется формула Шеннона для определения количества информации:

$$I = - \sum_{i=1}^n p_i \log_2 p_i.$$

$$I = -(p_2 \cdot \text{Log}_2 p_2 + p_3 \cdot \text{Log}_2 p_3 + p_4 \cdot \text{Log}_2 p_4 + p_5 \cdot \text{Log}_2 p_5) = -(1/8 \cdot \text{Log}_2 (1/8) + 1/8 \cdot \text{Log}_2 (1/8) + 3/14 \cdot \text{Log}_2 (3/14) + 5/32 \cdot \text{Log}_2 (5/32)) \approx 0.375 + 0.375 + 0.48 + 0.42 \approx 1.65$$
 (бит)

**Пример 2.** Монету бросают четыре раза. Определите, сколько бит информации несет сообщение о том, что орел выпал три раза.

**Решение:**

Для определения количества информации в битах для сообщения о том, что орел выпал три раза используется формула:

$$I_A = \text{Log}_2 (1/ P_A)$$

$P_A$  – вероятность того, что орел выпал три раза из четырех бросков

$P_A = ( \text{Количество сочетаний «три раза орел из 4-х бросков» } ) / ( \text{Количество комбинаций из 4-х подбрасываний} )$

**Количество сочетаний «три раза орел из 4-х бросков» определяется по формуле:**

$$C_n^k = \frac{n!}{k!(n-k)!}, \text{ где } C \text{ – число сочетаний из } n \text{ возможных событий по } k \text{ раз.}$$

Для данного примера  $n = 4, k=3$ , тогда:

$$C_4^3 = \frac{4!}{3!(4-3)!} = 4$$

**Количество комбинаций из 4-х подбрасываний =  $2^4 = 16$**

Тогда :

$$P_A = 4 / 16 ;$$

$$I_A = \text{Log}_2 (1/ P_A) = \text{Log}_2 (1/ (4/16)) = 2 \text{ бита}$$

## 2. Алфавитный подход

Алфавит рассматривается как «конечная последовательность». Для конечных последовательностей понятие вероятности совпадает с понятием частоты появления данного символа среди множества символов последовательности.

Формула Шеннона для случая N-символьного алфавита определяет *средний информационный вес одного символа* такого алфавита.

**Пример 1.** Частотный словарь русского языка - словарь вероятностей (частот) появления букв в произвольном тексте - приведен ниже. Определите, какое количество информации несет отдельная буква этого словаря?

Символ	Частота	Символ	Частота	Символ	Частота	Символ	Частота
о	0,090	в	0,035	я	0,018	ж	0,007
е, ё	0,072	к	0,028	ы, з	0,016	ю, ш	0,006
а, и	0,062	м	0,026	ь, ъ, б	0,014	ц, щ, э	0,003
т, н	0,053	д	0,025	ч	0,013	ф	0,002
с	0,045	п	0,023	й	0,012		
р	0,040	у	0,021	х	0,009		

**Решение:**

Применяется формула:  $I_A = \text{Log}_2 (1/ P_A)$ , где

$I_A$  – количество информации, которое несет отдельная буква алфавита;

$P_A$  – вероятность (частота) появления отдельной буквы алфавита.

Количество информации которое несет буква «и»:

$$I_{и} = \text{Log}_2 (1/ P_{и}) = \text{Log}_2 (1/ 0,062) \approx \mathbf{3,768 \text{ (бит)}}$$

Количество информации которое несет буква «ф»:

$$I_{ф} = \text{Log}_2 (1/ P_{ф}) = \text{Log}_2 (1/ 0,002) \approx \mathbf{8,97 \text{ (бит)}}$$

**Пример 2.** В алфавите, которым был закодирован текст, 8 символов:  $\vartheta, \Omega, \mathfrak{Z}, \wp, \&, \zeta, \Theta$ , «пробел». Подсчитано, что в сообщении (в сообщении присутствуют все символы алфавита) содержится 1000 знаков, из них указанные символы встречаются определенное количество раз:  $\vartheta=250, \mathfrak{Z}=100, \Omega=140, \wp=270, \&=100, \zeta=50, \Theta=80$ , «пробел»=10. Какой объем информации в переданном сообщении (алфавитный подход).

1. Подсчет частоты встречаемости каждого символа	2. Информационный вес каждого символа
$\vartheta=250/1000= 0,25$	$i_{\vartheta}=\log_2(1/0,25)\approx 2$
$\mathfrak{Z}=100/1000= 0,1$	$i_{\mathfrak{Z}}=\log_2(1/0,1)\approx 3,3$
$\Omega=140/1000= 0,14$	$i_{\Omega}=\log_2(1/0,14)\approx 2,8$
$\wp=270/1000= 0,27$	$i_{\wp}=\log_2(1/0,27)\approx$
$\&=100/1000= 0,1$	$i_{\&}=\log_2(1/0,1)\approx$
$\zeta=50/1000=0,05$	$i_{\zeta}=\log_2(1/0,05)\approx$
$\Theta=80/1000=0,08$	$i_{\Theta}=\log_2(1/0,08)\approx$
«пробел»=10/1000=0,001	«пробел»" = $\log_2(1/0,001)\approx$

3. Общий объем информации в закодированном сообщении: сумма произведений информационного веса каждого символа на количество повторений данного символа в тексте сообщений.

$$I=i_{\vartheta}*n_{\vartheta}+ i_{\mathfrak{Z}}*n_{\mathfrak{Z}}+\dots+ i_{\text{пробел}}*n_{\text{пробел}} .$$

**Задание №1**

Документация некоторого учреждения размещена в восьми комнатах. В каждой комнате находится 32 шкафа. Каждый шкаф имеет 8 полок. Определить количество информации, которое несет сообщение о том, что нужный документ находится в третьей комнате, в тринадцатом шкафу на пятой полке.

**Задание №2**

По результатам года обучения студент получил по предметам 100 оценок. Из них: 60 «отлично», 30 «хорошо», 8 «удовлетворительно» и 2 «неудовлетворительно».

Сообщение о получении какой оценки содержит меньше всего информации?

**Задание №3**

В течение часа к остановке подходят автобусы: №4 и №15 – 4 раза, №8 – 5 раз, №10 – 3 раза. Сравните количество информации в сообщениях: «*подошел автобус №4*» и «*подошел автобус №10*». Обоснуйте полученный ответ.

**Задание №4**

Известно, что в ящике лежат 60 шаров. Из них 12-черных, 25-желтых, 8-красных, остальные шары белого цвета. Сколько бит информации в сообщении о том, что из ящика случайным образом достали белый шар?

**Задание №5**

При проведении серии измерений некоторой величины  $A$ , прибор фиксировал значения «ниже нормы» 7 раз, выше нормы 11 раз, в пределах нормы - 1 раз. Сколько бит информации в сообщении о том, что прибор показал значение выше нормы?

**Задание №6**

Проведен ряд экспериментов, в которых симметричную монету бросают семь раз. Определите, сколько бит информации несет сообщение о том, что решка выпадала два раза.

**Задание №1**

Подсчитайте количество информации, приходящейся на один символ текста, соответствующему Вашему варианту.

Указания к выполнению задания:

- составьте таблицу, аналогичную таблице «Частотность букв русского алфавита» (см. ниже), определив вероятность каждого символа в тексте как отношение количества одинаковых символов каждого значения ко всему числу символов в тексте;
- подсчитайте количество информации, приходящейся на один символ по формуле Шеннона.
- задание выполните с использованием табличного процессора (Excel, OpenOffice Calc и т.п.), сохранив результат работы в соответствующем файле.

**Частотность букв русского языка**

i	Символ	P(i)	i	Символ	P(i)	i	Символ	P(i)
1	–	0.175	12	Л	0.035	23	Б	0.014
2	О	0.090	13	К	0.028	24	Г	0.012
3	Е	0.072	14	М	0.026	25	Ч	0.012
4	Ё	0.072	15	Д	0.025	26	Й	0.010
5	А	0.062	16	П	0.023	27	Х	0.009
6	И	0.062	17	У	0.021	28	Ж	0.007
7	Т	0.053	18	Я	0.018	29	Ю	0.006
8	Н	0.053	19	Ы	0.016	30	Ш	0.006
9	С	0.045	20	З	0.016	31	Ц	0.004
10	Р	0.040	21	Ь	0.014	32	Щ	0.003
11	В	0.038	22	Ъ	0.014	33	Э	0.003
						34	Ф	0.002

№ варианта	Текст
1	Информацию, рассматриваемую только с синтаксических позиций, обычно называют данными, так как в этой форме смысловая сторона не имеет значения. Эта форма способствует восприятию внешних структурных характеристик, то есть синтаксической стороны информации. Количественную, структурную сторону информации изучает синтактика, раздел семиотики — науки о знаках и знаковых системах.
2	<i>Семантическая (смысловая) адекватность</i> определяет степень соответствия образа объекта и самого объекта. Семантический аспект предполагает учет смыслового содержания информации. Эта форма служит для формирования понятий и представлений, выявления смысла, содержания информации, для ее обобщения.

№ варианта	Текст
3	<i>Прагматическая (потребительская) адекватность</i> отражает отношение информации и ее потребителя, соответствие информации цели управления, которая на ее основе реализуется. Прагматический аспект рассмотрения связан с ценностью, полезностью использования информации при выработке потребителем решения для достижения своей цели.
4	Под <i>качеством</i> информации понимается такая совокупность свойств, которая обуславливает ее способность удовлетворять определенные потребности людей. Основными потребительскими показателями качества информации являются: репрезентативность; содержательность; достаточность; доступность; актуальность; своевременность; точность; достоверность; устойчивость.
5	<i>Репрезентативность</i> информации связана с правильностью ее отбора и формирования в целях адекватного отражения свойств объекта. Важнейшее значение здесь имеют: правильность концепции, на базе которой сформулировано исходное понятие; обоснованность отбора существенных признаков и связей отображаемого явления. Нарушение репрезентативности информации приводит нередко к существенным ее погрешностям.
6	<i>Достаточность (полнота)</i> информации означает, что ее состав (набор показателей) минимален, но достаточен для принятия правильного решения. Понятие полноты информации связано с ее смысловым содержанием (семантикой) и прагматикой. Как неполная, то есть недостаточная для принятия правильного решения, так и избыточная информация снижает эффективность принимаемых пользователем решений.
7	<i>Доступность</i> информации восприятию пользователя обеспечивается выполнением соответствующих процедур ее получения и преобразования. Например, в информационной системе информация преобразуется к доступной и удобной для восприятия пользователем форме. Это достигается, в частности, и путем согласования ее семантической формы с тезаурусом пользователя.
8	<i>Актуальность</i> информации определяется степенью сохранения ценности информации для управления на момент использования и зависит от динамики изменения ее характеристик, а также от интервала времени, прошедшего с момента возникновения данной информации. <i>Своевременность</i> информации означает ее поступление не позже заранее назначенного момента времени, согласованного с временем решения поставленной задачи.
9	<i>Точность</i> информации определяется степенью близости получаемой информации к реальному состоянию объекта, процесса, явления и т. п. Для информации, отображаемой цифровым кодом, известны четыре классификационных понятия точности: <ul style="list-style-type: none"> <li>• формальная точность измеряется значением единицы младшего разряда числа;</li> <li>• реальная точность определяется значением единицы последнего разряда числа, верность которого гарантируется;</li> </ul>



№ варианта	Текст
	<ul style="list-style-type: none"> <li>• максимальная точность представляет собой точность, которую можно получить в конкретных условиях функционирования системы;</li> <li>• необходимая точность определяется функциональным назначением показателя.</li> </ul>
10	<p><i>Достоверность</i> информации определяется ее свойством отражать реально существующие объекты с необходимой точностью. Измеряется достоверность информации доверительной вероятностью необходимой точности, то есть вероятностью того, что отображаемое информацией значение параметра отличается от истинного значения этого параметра в пределах необходимой точности.</p>
11	<p><i>Устойчивость</i> информации отражает ее способность реагировать на изменения исходных данных без нарушения необходимой точности. Устойчивость информации, как и репрезентативность, обусловлена выбранной методикой ее отбора и формирования.</p>
12	<p>Наиболее эффективным средством организации информационных процессов является <i>информационная система</i>, оснащенная средствами ввода, поиска, размещения, обработки и выдачи информации. Наличие таких средств — главная особенность информационных систем, отличающих их от простых скоплений информационных материалов.</p>
13	<p>Путем кодирования производится отображение одного набора знаков в другой набор знаков. Отображаемый набор знаков называется исходным алфавитом, а набор знаков, который используется для отображения, — кодовым алфавитом. Для построения кода применяются как отдельные символы кодового алфавита, так и их комбинации.</p>
14	<p><i>Данные</i> могут рассматриваться как признаки или записанные наблюдения, которые по каким-то причинам не используются, а только хранятся. В том случае, если появляется возможность использовать эти данные для уменьшения неопределенности знаний о чем-либо, данные превращаются в информацию.</p>

## ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

1. Макарова Н. В., Волков В. Б. Информатика: учебник для вузов. СПб.: Питер, 2011. 576 с.
2. Могилев А. В. Практикум по информатике: учеб. пособие для студ. высш. учеб. заведений / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. М. : Академия, 2008. 608 с.
3. «Информатика»: сайт. URL: [http://informatika.ru/lectures/izmerenie\\_informacii.html](http://informatika.ru/lectures/izmerenie_informacii.html).
4. «Экономико-математический словарь»  
URL: [https://economic\\_mathematics.academic.ru/4075](https://economic_mathematics.academic.ru/4075).

## КОДИРОВАНИЕ ИНФОРМАЦИИ

---

В процессе преобразования информации из одной формы представления (знаковой системы) в другую осуществляется кодирование.

Кодирование – это представление сигнала в определенной форме, удобной или пригодной для последующего использования сигнала. Говоря строже, это правило, описывающее отображение одного набора знаков в другой набор знаков. Тогда отображаемый набор знаков называется исходным алфавитом, а набор знаков, который используется для отображения, – кодовым алфавитом, или алфавитом для кодирования. При этом кодированию подлежат как отдельные символы исходного алфавита, так и их комбинации. Аналогично для построения кода используются как отдельные символы кодового алфавита, так и их комбинации [3].

В канале связи сообщение, составленное из символов (букв) одного алфавита, может преобразовываться в сообщение из символов (букв) другого алфавита. Правило, описывающее однозначное соответствие букв алфавитов при таком преобразовании, называют кодом. Саму процедуру преобразования сообщения называют перекодировкой. Подобное преобразование сообщения может осуществляться в момент поступления сообщения от источника в канал связи (кодирование) и в момент приема сообщения получателем (декодирование)[2].

Цель кодирования – более сжатое, компактное представление сообщения.

С кодированием связаны понятия: *код, декодирование*.

Код – правило соответствия набора знаков одного множества  $X$  знакам другого множества  $Y$ .

Если каждому символу  $X$  при кодировании соответствует отдельный знак  $Y$ , то это кодирование.

Если для каждого символа из  $Y$  однозначно отыщется по некоторому правилу его прообраз в  $X$ , то это правило называется *декодированием* [5].

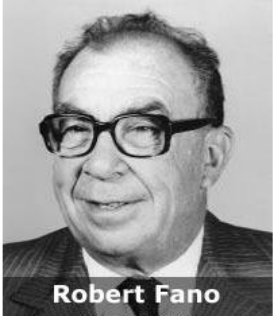
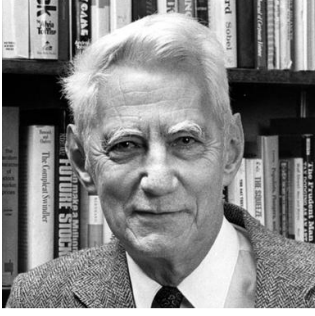
Различают кодирование:

- числовой информации:
  - целые числа;
  - вещественные числа;
- нечисловой информации:
  - текстовой;
  - графической;
  - звуковой;
  - и другие.

Существуют различные способы кодирования информации. Двоичное кодирование – один из распространенных способов представления информации.

## КОД ШЕННОНА – ФАНО

Клод Шеннон и Роберт Фано независимо друг от друга в 1948-1949 гг. предложили способ построения кода, основанный на выполнении условия равной вероятности символов 0 и 1 в закодированном сообщении.

 <b>Robert Fano</b>	Итальяно-американский учёный в области информатики, независимо от Клода Шеннона изобрел ранний алгоритм сжатия информации		Американский инженер, криптоаналитик и математик, считается «отцом информационного века», является основателем теории информации
Роберт Марио Фанс		Клод Элвуд Шеннон	

**Пример 1.** Построение кода Шеннона–Фано и расчет *средней длины, эффективности и избыточности кода.*

*Дано:* сообщения  $x_j$  с вероятностями:

$$x_1 = 0,09; x_2 = 0,4; x_3 = 0,11; x_4 = 0,23; x_5 = 0,1; x_6 = 0,07.$$

*Необходимо:* 1) построить код Шеннона–Фано; 2) рассчитать: среднюю длину, эффективность и избыточность кода.

**Решение:**

**1. Постройте Код Шеннона–Фано:**

Буква	Вероятности	Разряды				Код Шеннона–Фано
$x_2$	<b>0,4</b>	<b>0</b>				<b>0</b>
$x_4$	<b>0,23</b>		<b>0</b>	<b>0</b>		<b>100</b>
$x_3$	<b>0,11</b>			<b>1</b>		<b>101</b>
$x_5$	<b>0,1</b>			<b>0</b>		<b>110</b>
$x_1$	<b>0,09</b>				<b>0</b>	<b>1110</b>
$x_6$	<b>0,07</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1111</b>

## 2. Рассчитайте среднюю длину кода.

### 2.1. Определите длину кодового слова:

Буква	Вероятности	Разряды				Код Шеннона–Фано	Длина кодового слова
X <sub>2</sub>	0,4	0				0	1
X <sub>4</sub>	0,23	0	0	0		100	3
X <sub>3</sub>	0,11			1		101	3
X <sub>5</sub>	0,1			0		110	3
X <sub>1</sub>	0,09	1	1	1	0	1110	4
X <sub>6</sub>	0,07				1		1111

### 2.2. Рассчитайте среднюю длину кода по формуле:

$$L_{cp} = \sum_{i=1}^M P(x_i) * n_i$$

, где

$P(x_i)$  – вероятность появления элементов из алфавита  $M$ ,

$n_i$  – длина кодового слова, соответствующего элементу  $x_i$

$$L_{cp} = \sum_{i=1}^M P(x_i) * n_i = 0,09 \cdot 4 + 0,4 \cdot 1 + 0,11 \cdot 3 + 0,23 \cdot 3 + 0,1 \cdot 3 + 0,07 \cdot 4 \approx 2,36 \text{ бит.}$$

## 3. Рассчитайте эффективность кода.

*Эффективность кода ( $\chi$ ) определяется по формуле:*

$\chi = H / L_{cp}$  , где  $H$  – энтропия источника сообщения;  $L_{cp}$  – средняя длина кода.

### 3.1. Расчет энтропии источника сообщения:

$H$  – энтропия источника сообщения (среднее значение количества информации, приходящееся на одно элементарное сообщение).

Для данного примера количество сообщений:  $m=6$ .

$$H = -\sum_{i=1}^m p_i \log_2 p_i = -\sum_{i=1}^m p_{X_i} \log_2 p_{X_i}$$

Буква	Вероятности	Разряды				Код Шеннона–Фано	Длина кодового слова	$H = -p \cdot \log_2(p)$
X <sub>2</sub>	0,4	0				0	1	-0,52877
X <sub>4</sub>	0,23	1	1	0		100	3	-0,48767
X <sub>3</sub>	0,11			1		101	3	-0,35029
X <sub>5</sub>	0,1			0		110	3	-0,33219
X <sub>1</sub>	0,09				0	1110	4	-0,31265
X <sub>6</sub>	0,07			1	1	1	1	1111
<b>Энтропия (H) ≈</b>								<b>2,28013</b>

Энтропия  $H(x) \approx 2,28$  бит (для данной кодовой таблицы).

3.2. Средняя длина кода ( $L_{cp}$ ) рассчитана в пункте 2.2;

Эффективность кода:  $\chi = H / L_{cp} = 2,28 / 2,36 \approx 0,966102$ .

4. Рассчитайте избыточность кода.

Избыточность сообщения ( $D$ ) определяется:  $D = 1 - H/H_{max}$ ;

$H \approx 2,28013$  (см. пункт 3.1.)

$H_{max}$  -?

Для алфавита из 8-х букв ( $m=8$ ), максимальная энтропия рассчитывается:  $H_{max} = \log_2 m = \log_2 8 = 3$ .

В данном примере кодовое слово состоит из 6 символов ( $x_1, x_2, x_3, x_4, x_5, x_6$ ), тогда  $m=6$ .

Следовательно,  $H_{max}$  рассчитывается:  $H_{max} = \log_2 6 \approx 2,584962$ .

Избыточность  $D$  рассчитывается:  $D = 1 - H/H_{max} = 1 - 2,28013 / 2,584962 \approx 0,12$ .

Эффективность кода:  $\chi \approx 0,97$ .

Средняя длина кода:  $L_{cp} \approx 2,28$  бит.

Избыточность кода:  $D \approx 0,12$ .

**Задание №1**

Для каждого из представленных сообщений:

- постройте код Шеннона–Фано;
- вычислите среднюю длину кодового слова, эффективность и избыточность кода.

Все построения и расчеты выполните с использованием табличного процессора.

№ варианта	Задание
1	Дано множество из нескольких сообщений $X_i$ с вероятностями: 0,02; 0,51; 0,04; 0,015; 0,015; 0,37; 0,015; 0,015
2	Дано множество из нескольких сообщений $A_i$ с вероятностями: 0,42; 0,298; 0,04; 0,012; 0,08; 0,06; 0,09
3	Дано множество из нескольких сообщений $C_i$ с вероятностями: 0,07; 0,348; 0,03; 0,002; 0,05; 0,08; 0,019; 0,02; 0,063; 0,3; 0,018
4	Дано множество из нескольких сообщений $C_i$ с вероятностями: 0,288; 0,08; 0,41; 0,002; 0,07; 0,05; 0,08; 0,07
5	Дано множество из нескольких сообщений $B_i$ с вероятностями: 0,11; 0,01; 0,2; 0,057; 0,06; 0,013; 0,08; 0,027; 0,02; 0,063; 0,36
6	Дано множество из нескольких сообщений $D_i$ с вероятностями: 0,36; 0,04; 0,2; 0,049; 0,06; 0,04; 0,087; 0,023; 0,15; 0,04
7	Дано множество из нескольких сообщений $Q_i$ с вероятностями: 0,01; 0,01; 0,2; 0,02; 0,012; 0,045; 0,078; 0,5; 0,105; 0,01; 0,01
8	Дано множество из нескольких сообщений $Z_i$ с вероятностями: 0,079; 0,004; 0,07; 0,1; 0,033; 0,01; 0,014; 0,39; 0,3
9	Дано множество из нескольких сообщений $T_i$ с вероятностями: 0,041; 0,1; 0,016; 0,01; 0,02; 0,06; 0,05; 0,7; 0,003
10	Дано множество из нескольких сообщений $R_i$ с вероятностями: 0,008; 0,05; 0,007; 0,071; 0,023; 0,074; 0,069; 0,091; 0,508; 0,099
11	Дано множество из нескольких сообщений $W_i$ с вероятностями: 0,03; 0,063; 0,055; 0,605; 0,015; 0,084; 0,069; 0,047; 0,025; 0,007
12	Дано множество из нескольких сообщений $F_i$ с вероятностями: 0,038; 0,1; 0,01; 0,108; 0,207; 0,109; 0,135; 0,208; 0,079; 0,006
13	Дано множество из нескольких сообщений $V_i$ с вероятностями: 0,05; 0,03; 0,01; 0,08; 0,011; 0,651; 0,017; 0,044; 0,079; 0,006; 0,022
14	Дано множество из нескольких сообщений $V_i$ с вероятностями: 0,4; 0,004; 0,02; 0,02; 0,007; 0,371; 0,03; 0,016; 0,032; 0,083; 0,017

**Задание №1**

*Код Хаффмана*

**1.1. Изучить материал, представленный ниже**

Алгоритм Хаффмана – это алгоритм оптимального префиксного (ни одно кодовое слово не является префиксом другого) кодирования алфавита с минимальной избыточностью. Алгоритм был разработан в 1952 г. студентом Массачусетского технологического института Дэвидом Хаффманом. Сжатие данных по Хаффману применяется при сжатии фотоизображений, видеоизображений (JPEG, MPEG), а также в архиваторах PKZIP , ARJ и др.

**Пример №1.** Пример построения кода Хаффмана:

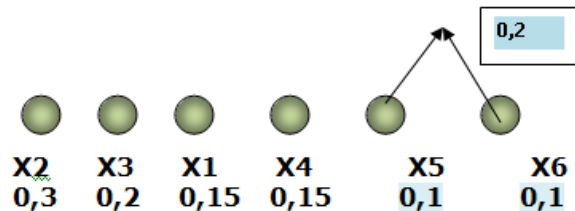
Дано: последовательность символов  $X_1, X_2, X_3, X_4, X_5, X_6$  с соответствующими вероятностями  $0,15; 0,3; 0,2; 0,15; 0,1; 0,1$ .  
 Необходимо: построить код Хаффмана.

**Решение:**

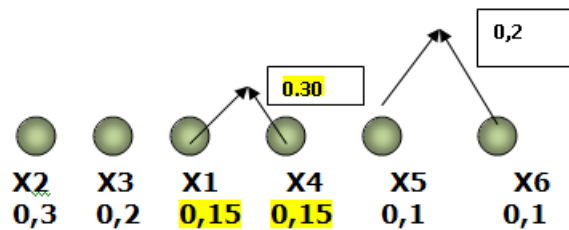
- Упорядочить все элементы сообщения по невозрастанию их вероятностей (частот встречаемости). Эти элементы образуют концевые вершины дерева, называемые также листьями.

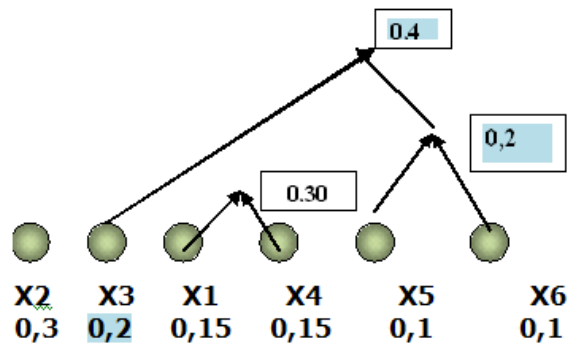
$X_2 \quad X_3 \quad X_1 \quad X_4 \quad X_5 \quad X_6$   
 $0,3 \quad 0,2 \quad 0,15 \quad 0,15 \quad 0,1 \quad 0,1$

- Выбрать две из них с **наименьшими вероятностями**. Построить новую вершину, одна ветвь которой идет к одному элементу, а другая — к другому. Этой новой вершине припишем вероятность, **равную сумме вероятностей** вершин, с которыми она соединена.

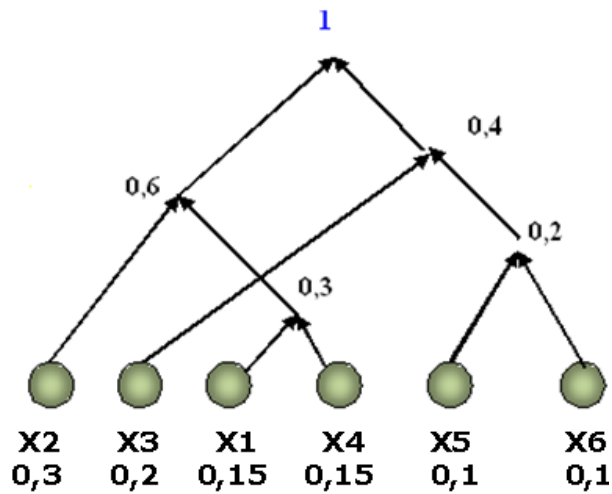


- Рассмотреть все оставшиеся концевые вершины и только что построенную. Выбрать из них две с минимальными вероятностями и соединить их с новой вершиной, вероятность которой будет равна сумме вероятностей объединяемых вершин:

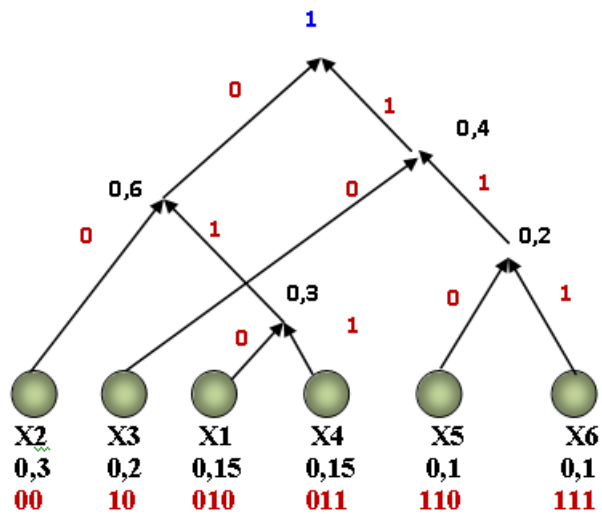




- Прodelать аналогичные действия до тех пор, пока не останется ни одной пары вершин, которые можно было бы объединить. Результат – полученное дерево.



- Разметить ветви, выходящие из вершин: одной из них припишем 1, а другой 0. Сделаем это для всех пар ветвей.





- Последовательность нулей и единиц, которую можно «собрать», двигаясь от корня дерева к концевой вершине, образует кодовое слово для соответствующего элемента.

1.2. Для Задания №1 из лабораторной работы постройте код Хаффмана, согласно Вашему варианту.

1.3. Определите эффективность построенного кода.

1.4. Сравните эффективность кода Шеннона и эффективность кода Хаффмана.

**Задание №2.** Рассмотрите следующие примеры:

2.1. Пример построения кода Хаффмана для сообщения «Код Хаффмана».

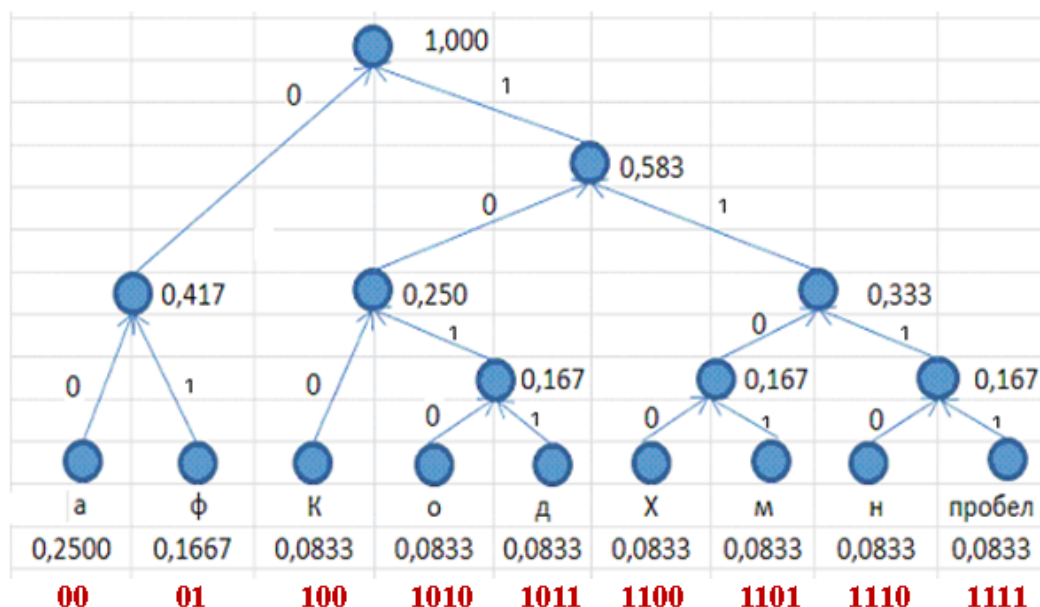
- определяется количество символов в сообщении:  $S=12$
- определяется алфавит фразы:  $M=9$  (включая пробел)

Алфавит	К	о	д	Х	а	ф	м	н	пробел
---------	---	---	---	---	---	---	---	---	--------

- рассчитываются количество вхождений каждого символа в фразу ( $N$ ) и вероятности символов ( $p$ ):

Алфавит	К	о	д	Х	а	ф	м	н	пробел
Кол-во вхождений символа в сообщении N	1	1	1	1	3	2	1	1	1
Вероятность $p=N_i/S$	0.083	0.083	0.083	0.083	0.25	0.167	0.083	0.083	0.083

- строится код Хаффмана:



Код сообщения «Код Хаффмана»: 100 1010 1011 1111 1100 00 01 01 1101 00 1110 00

**2.2. Пример определения коэффициента сжатия** относительно равномерного кодирования (5 бит на один символ).

$$K_{CЖ} = (L_{\text{равномерного кода}}) / (L_{\text{Хаффмана}})$$

- длина кода при равномерном кодировании:

$$L_{\text{равномерного кода}} = S \cdot i, \text{ где}$$

S – количество символов в строке; i – количество информации в одном символе (бит)

В данном примере, S=12, i=5 ;  $L_{\text{ср равномерный код}} = S \cdot i = 12 \cdot 5 = 60$  бит.

- длина кода при кодировании кодом Хаффмана:

$$L_{\text{Хаффман}} = L_{\text{символа}} \cdot N, \text{ где}$$

$L_{\text{символа}}$  – длина кода символа; N – количество вхождений символа в сообщение.

Символы	а	ф	к	о	д	х	м	н	пробел
Кол-во вхождений символа в сообщение N	3	2	1	1	1	1	1	1	1
Код	00	01	100	1010	1011	1100	1101	1110	1111
Длина кода символа $L_{\text{символа}}$	2	2	3	4	4	4	4	4	4

Для данного примера:

$$L_{\text{Хаффман}} = L_{\text{символа}} \cdot N = 2 \cdot 3 + 2 \cdot 2 + 4 \cdot 1 + 4 \cdot 1 + 4 \cdot 1 + 4 \cdot 1 + 4 \cdot 1 + 4 \cdot 1 + 4 \cdot 1 = 38 \text{ бит}$$

$$K_{CЖ} = (L_{\text{равномерного кода}}) / (L_{\text{Хаффмана}}) = 60 / 38 \approx 1,579;$$

Коэффициента сжатия относительно равномерного кодирования:  $K_{CЖ} \approx 1,579$ .

**2.3. Пример расчета дисперсии кода.**

Дисперсия показывает существующее отклонение длины кода символа от средней длины кода. Наилучшим считается код с наименьшей дисперсией кода.

$$\delta = \sum_{i=1}^M p_i (L_i - L_{\text{ср}})^2, \text{ где}$$

M – количество символов алфавита, с помощью которого записано сообщение;

$P_i$  – вероятность появления символа алфавита;

$L_i$  – длина кода символа (количество бит в коде символа);

$L_{\text{ср}}$  – средняя длина кода Хаффмана:

$$L_{\text{ср}} = \sum_{i=1}^M P_i * L_i$$

Для данного примера:

$M=9$ ;

$$L_{cp} = \sum_{i=1}^M p_i * L_i = 0,25 \cdot 2 + 0,1667 \cdot 2 + 0,0833 \cdot 3 + 0,0833 \cdot 4 + 0,0833 \cdot 4 + 0,0833 \cdot 4 + 0,0833 \cdot 4 + 0,0833 \cdot 4 + 0,0833 \cdot 4 \approx 3,09$$

$$\begin{aligned} \delta &= \sum_{i=1}^i p_i (L_i - L_{\tilde{n}\delta})^2 = 0,25 \cdot (2 - 3,09)^2 + 0,1667 \cdot (2 - 3,09)^2 + 0,0833 \cdot (3 - 3,09)^2 + \\ &0,0833 \cdot (4 - 3,09)^2 + 0,0833 \cdot (4 - 3,09)^2 + 0,0833 \cdot (4 - 3,09)^2 + 0,0833 \cdot (4 - 3,09)^2 + 0,0833 \cdot \\ &(4 - 3,09)^2 + 0,0833 \cdot (4 - 3,09)^2 = \\ &= 0,25 \cdot (2 - 3,09)^2 + 0,1667 \cdot (2 - 3,09)^2 + 0,0833 \cdot (3 - 3,09)^2 + 6 \cdot (0,0833 \cdot (4 - 3,09)^2) \\ &\approx 0,9786; \end{aligned}$$

Дисперсия кода:  $\delta \approx 0,9786$

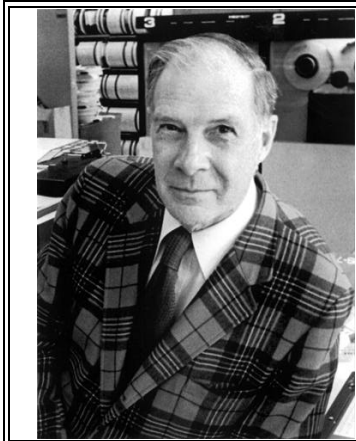
**2.4. Постройте код Хаффмана для сообщения, состоящего из своей Фамилии, Имени и Отчества.**

**2.5. Рассчитайте коэффициент сжатия относительно кодировки «Юникод» (2 байта на один символ).**

**2.6. Рассчитайте дисперсию кода.**

*Все построения и расчеты выполните с использованием табличного процессора.*

## КОД ХЕММИНГА



Американский математик, работы которого в сфере теории информации оказали существенное влияние на компьютерные науки и телекоммуникации.

Ричарда Хэмминга называют «гением одной идеи». Он сформулировал ее в 1950 г. в своей научной статье, посвященной кодам для коррекции ошибок. Статья содержала конструкцию блочного кода, корректирующего одиночные ошибки, которые возникают при передаче сообщений

Ричард Уэсли Хэмминг

- k** – количество информационных разрядов;
- r** – количество проверочных (контрольных) разрядов;
- n** – количество разрядов в коде Хемминга (длина кода Хемминга).
- n = k + r**, где
- k** – определяется из двоичного кода;
- r ≥ Log<sub>2</sub> (k + 1)** (значение r округляется в «большую сторону»)

**Пример 1.** Построение кода Хемминга для передачи числа  $2_{10}$ .

1. *Переведите число 2 в двоичную систему счисления  $2_{10} = 10_2$ .*
2. *Определите число информационных символов в коде: k = 2 (1 и 0).*
3. *Определите число контрольных символов r в коде:*

$k = 2; r \geq \text{Log}_2 (k + 1) = \text{Log}_2 (2 + 1) \approx 1,58 = 2. ; r \geq 2; r = 3$  (см. табл. 1).

4. *Определите длину кода Хемминга n, используя таблицу 1.*  
r = 3; k = 2; n = 5.

Таблица 1

Соотношение n, r и k для кода Хэмминга

<i>n</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
<i>k</i>	0	0	1	1	2	3	4	4	5	6	7	8	9	10	11	11
<i>r</i>	1	2	2	3	3	3	3	4	4	4	4	4	4	4	4	5

5. *Постройте таблицу 2.*

Таблица 2

Проверочные позиции					
1	2	3	4	5	Разряды кода Хемминга
					Назначение разрядов
					Значение разряда

6. Заполните таблицу, определив позиции и значения разрядов:

6.1. Определите позиции контрольных (r) и информационных (k) коэффициентов в коде.

Для r позиция определяется значениями степени двойки:  $2^i$ , где  $i = 0, 1, 2, 3 \dots$   
1, 2, 4, 8, 16, ...

для  $r=3$  позиции контрольных коэффициентов: 1, 2, 4.

Для k в качестве позиций – остальные разряды: 3, 5 (всего разрядов:  $n=5$ ):

Таблица 3

Проверочные позиции					
1	2	3	4	5	Разряды кода Хемминга
r1	r2	k1	r3	k2	Назначение разрядов
					Значение разряда

6.2. Заполните значения информационных разрядов (k1, k2):

Проверочные позиции					
1	2	3	4	5	Разряды кода Хемминга
r1	r2	k1	r3	k2	Назначение разрядов
		1		0	Значение разряда

6.3. Заполните значения контрольных разрядов.

Каждый контрольный разряд «проверяет» информационные разряды. Значения контрольных символов определяют по правилу: если число единиц на проверочных позициях четное, то значение контрольного коэффициента равно 0 ( $r = 0$ ), в противном случае – 1 ( $r = 1$ ).

Используя табл. 4, находим значения контрольных разрядов:

Таблица 4

Порядковый номер контрольного разряда	Позиция контрольного разряда	Проверочные разряды
r1	1	1, 3, 5, 7, 9, 11, 13, ...
r2	2	2, 3, 6, 7, 10, 11, 14, 15, 18, 19, 22, 23, ...
r3	4	4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23, 28, 29, 30, 31, ...
r4	8	8, 9, 10, 11, 12, 13, 14, 15, 24, 25, 26, 27, 28, 29, 30, 31, 40, 41, ...

$r1 = (r3, r5) = (1, 0) = 1$  (число единиц на проверочных позициях = 1, нечетное)

$r2 = r3 = 1$  (число единиц на проверочных позициях = 1, нечетное)

$r3 = r5 = 0$  (число единиц на проверочных позициях = 0 – единиц нет, ноль четное число)

В результате получим таблицу:

Таблица 5

Проверочные позиции					
1	2	3	4	5	Разряды кода Хемминга
<b>r1</b>	<b>r2</b>	<b>k1</b>	<b>r3</b>	<b>k2</b>	Назначение разрядов
<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	Значение разряда

**Полученный код Хемминга: 11100 – для передачи числа 2**

**Задание №1.** Постройте код Хемминга.

№ Варианта	Числа	Сообщение
1	$A = 35_{10}$ ; $D = 57_8$ ; $E = 123_{16}$	do
2	$A = 23_{10}$ ; $D = 47_8$ ; $E = 111_{16}$	go
3	$A = 21_{10}$ ; $D = 37_8$ ; $E = 121_{16}$	as
4	$A = 33_{10}$ ; $D = 57_8$ ; $E = 122_{16}$	or
5	$A = 38_{10}$ ; $D = 51_8$ ; $E = 130_{16}$	an
6	$A = 39_{10}$ ; $D = 36_8$ ; $E = 109_{16}$	my
7	$A = 35_{10}$ ; $D = 17_8$ ; $E = 115_{16}$	me
8	$A = 32_{10}$ ; $D = 45_8$ ; $E = 107_{16}$	so
9	$A = 31_{10}$ ; $D = 37_8$ ; $E = 105_{16}$	of
10	$A = 22_{10}$ ; $D = 14_8$ ; $E = 112_{16}$	at
11	$A = 27_{10}$ ; $D = 51_8$ ; $E = 120_{16}$	up
12	$A = 20_{10}$ ; $D = 16_8$ ; $E = 102_{16}$	he
13	$A = 21_{10}$ ; $D = 31_8$ ; $E = 117_{16}$	we
14	$A = 25_{10}$ ; $D = 43_8$ ; $E = 110_{16}$	to

Для определения кода символов, используйте часть стандартной таблицы кодов ASCII представленную ниже.

Часть стандартной таблицы кодов ASCII

	32	00100000	8	56	00111000	P	80	01010000	h	104	01101000
!	33	00100001	9	57	00111001	Q	81	01010001	i	105	01101001
"	34	00100010	:	58	00111010	R	82	01010010	j	106	01101010
#	35	00100011	;	59	00111011	S	83	01010011	k	107	01101011
\$	36	00100100	<	60	00111100	T	84	01010100	l	108	01101100
%	37	00100101	=	61	00111101	U	85	01010101	m	109	01101101
&	38	00100110	>	62	00111110	V	86	01010110	n	110	01101110
'	39	00100111	?	63	00111111	W	87	01010111	o	111	01101111
(	40	00101000	@	64	01000000	X	88	01011000	p	112	01110000
)	41	00101001	A	65	01000001	Y	89	01011001	q	113	01110001
*	42	00101010	B	66	01000010	Z	90	01011010	r	114	01110010
+	43	00101011	C	67	01000011	[	91	01011011	s	115	01110011
,	44	00101100	D	68	01000100	\	92	01011100	t	116	01110100
-	45	00101101	E	69	01000101	]	93	01011101	u	117	01110101
.	46	00101110	F	70	01000110	^	94	01011110	v	118	01110110
/	47	00101111	G	71	01000111	_	95	01011111	w	119	01110111
0	48	00110000	H	72	01001000	`	96	01100000	x	120	01111000
1	49	00110001	I	73	01001001	a	97	01100001	y	121	01111001
2	50	00110010	J	74	01001010	b	98	01100010	z	122	01111010
3	51	00110011	K	75	01001011	c	99	01100011	{	123	01111011
4	52	00110100	L	76	01001100	d	100	01100100		124	01111100
5	53	00110101	M	77	01001101	e	101	01100101	}	125	01111101
6	54	00110110	N	78	01001110	f	102	01100110	~	126	01111110
7	55	00110111	O	79	01001111	g	103	01100111	□	127	01111111



Задания для самостоятельной работы студента (СРС)

**Указания к выполнению:**

- для выполнения **Задания №2** воспользуйтесь *частью стандартной таблицы кодов ASCII*, представленной выше.
- для выполнения **Задания №3** изучите материал *«Код Хемминга. Исправление ошибки в коде»*.

№ Варианта	Задание №1	Задание №2	Задание №3
	Постройте код Хемминга для чисел	Постройте код Хемминга для сообщения	Определите, в каком разряде кода Хемминга находится ошибка, исправьте ошибку и восстановите переданное число. Результат представьте в двоичной и десятичной системах счисления
1	100D	cat	101001100110110
2	32 O	bit	111100111011010
3	10 H	talk	111101111011011
4	561 O	red	111101111111011
5	1A2 H	next	100001111011011
6	109 D	dog	100000000110111
7	462 O	add	101001000110111
8	B10 H	big	100100010110101
9	187 D	bad	101111000110101
10	222 O	air	101111000110100
11	12C H	set	101111010110100
12	176 D	else	110001010110100
13	1001 O	lot	101111000110111
14	64D H	top	111001000110100

**Материал для самостоятельного изучения. «Код Хемминга. Исправление ошибки в коде».**

**Пример 1.** По каналу связи был передан следующий код: **011100111011010** с ошибкой в одном из разрядов.

Определите, в каком разряде кода Хемминга находится ошибка. Исправьте ошибку. Определите число, которое было передано (с учетом исправлений).

**Решение:**

**1. Определите количество разрядов в коде:**

Разряды кода Хемминга	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Исходный код	0	1	1	1	0	0	1	1	1	0	1	1	0	1	0	

**n=15**

2. По таблице 1 определите  $k$  и  $r$ : для  $n = 15 \rightarrow k = 11 \quad r=4$

Таблица 1 Соотношение  $n, r$  и  $k$  для кода Хэмминга можно представить в виде таблицы:

$n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$k$	0	0	1	1	2	3	4	4	5	6	7	8	9	10	11	11
$r$	1	2	2	3	3	3	3	4	4	4	4	4	4	4	4	5

3. Определите разряд, в котором произошел сбой:

• По таблице 2 определите значения контрольных коэффициентов:

$r_1 = (r_1, r_3, r_5, r_7, r_9, r_{11}, r_{13}, r_{15}) = (0, 1, 0, 1, 1, 1, 0, 0) \rightarrow 0$  (число единиц = 4, четное)  
 $r_2 = (r_2, r_3, r_6, r_7, r_{10}, r_{11}, r_{14}, r_{15}) = (1, 1, 0, 1, 0, 1, 1, 0) \rightarrow 1$  (число единиц = 5, нечетное)  
 $r_3 = (r_4, r_5, r_6, r_7, r_{12}, r_{13}, r_{14}, r_{15}) = (1, 0, 0, 1, 1, 0, 1, 0) \rightarrow 0$   
 $r_4 = (r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14}, r_{15}) = (1, 1, 0, 1, 1, 0, 1, 0) \rightarrow 1$   
 **$r_1=0; r_2=1; r_3=0; r_4=1;$**

Таблица 2 значения контрольных коэффициентов

Порядковый номер контрольного разряда	Позиция контрольного разряда	Проверочные разряды
$r_1$	1	1, 3, 5, 7, 9, 11, 13, ...
$r_2$	2	2, 3, 6, 7, 10, 11, 14, 15, 18, 19, 22, 23, ...
$r_3$	4	4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23, 28, 29, 30, 31, ...
$r_4$	8	8, 9, 10, 11, 12, 13, 14, 15, 24, 25, 26, 27, 28, 29, 30, 31, 40, 41, ...

• Последовательность  $r_4, r_3, r_2, r_1$  дает двоичное число  $1010_2 = 10_{10} \rightarrow$  ошибка в **десятом** разряде.

4. Исправление ошибки - инвертирование ошибочного разряда:

Разряды кода Хемминга	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Исходный код	0	1	1	1	0	0	1	1	1	0	1	1	0	1	0	
Отредактированный код	0	1	1	1	0	0	1	1	1	1	1	1	0	1	0	

5. Определите число в двоичной системе счисления:

• Определите позиции контрольных разрядов:  $r_1=1; r_2=2; r_3=4; r_4=8.$

Разряды кода Хемминга	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Исходный код	0	1	1	1	0	0	1	1	1	0	1	1	0	1	0	
Отредактированный код	0	1	1	1	0	0	1	1	1	1	1	1	0	1	0	

- Выпишите значения из оставшиеся разрядов 3, 5, 6, 7, 9 - 15 – это значения *информационных разрядов*:

Разряды кода Хемминга	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Отредактированный код	0	1	1	1	0	0	1	1	1	1	1	1	0	1	0	

**1 0 0 1 1 1 1 1 0 1 0<sub>2</sub>**

**6. Переведите закодированное число** из двоичной системе счисления в 10-ую систему счисления:

$$1 0 0 1 1 1 1 1 0 1 0_2 = 1274_{10}$$

### ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

---

1. Лидовский В.И. Теория информации. М., «Высшая школа», 2002 г. 120 с.
2. Могилев А. В., Пак Н. И., Хеннер Е. К. Информатика: учебн. пособие для вузов. М.: Академия, 2004.
3. Кодирование информации (двоичные коды). Березюк Н. Т., Андрущенко А. Г., Мощицкий С.С. и др. Харьков, «Вища школа», 1978.
4. Беляев М.А., Лысенко В.В., Малинина Л.А. Основы информатики: учебник для вузов. URL: <http://wm-help.net/lib/b/book/120467185/6/>
5. Национальный открытый университет. «Кодирование и шифрование информации». URL: <https://www.intuit.ru/studies/courses/108/108/lecture/3143>.

## РАБОТА ПРОЦЕССОРА

---

Главным элементом персонального компьютера является микропроцессор, или центральный процессор – CPU (Central Processing Unit). Микропроцессор выполняет вычисления и обработку данных. Некоторые математические операции выполняет сопроцессор и является самой дорогостоящей микросхемой компьютера. Процессор выполняет операции над данными – операндами. Возможности использования ЭВМ определяются набором операций, которые может выполнять процессор. Этот набор операций называется системой команд. Ее объем является одной из главных характеристик ЭВМ. Для микроЭВМ система команд может содержать около сотни и более различных команд.

15 ноября 2016 года исполнилось 45 лет, как появился первый микропроцессор. За это время быстродействие процессора увеличилось более чем в 19 000 раз [5].

Основными характеристиками процессора считаются быстродействие и разрядность. Быстродействие измеряется в Гигагерцах (ГГц); 1 ГГц равен миллиарду тактов в секунду. Разрядностью процессора называется количество бит в обрабатываемых им числах, записанных в двоичной системе счисления. Эта техническая характеристика процессора является одной из самых важных, потому что определяет его быстродействие [6].

## КОМАНДЫ ПРОЦЕССОРА

Регистры процессора:



Адресация регистров общего назначения:



Семейство Intel. Регистры данных:

Регистры	Назначение
<b>AX</b>	Аккумулятор
<b>BX</b>	База
<b>CX</b>	Счетчик
<b>DX</b>	Данные

Для хранения АДРЕСОВ загружаемых данных существуют *сегментные* регистры и регистры *смещения*.

Регистры	Обозначение регистра	Название
<i>СЕГМЕНТНЫЕ РЕГИСТРЫ</i>	<b>CS</b>	Сегмент кода (программы)
	<b>DS</b>	Сегмент данных
	<b>SS</b>	Сегмент стека
	<b>ES</b>	Дополнительный сегмент
<i>РЕГИСТРЫ СМЕЩЕНИЯ</i>	<b>IP</b>	Указатель команд
	<b>SP</b>	Указатель стека
	<b>BP</b>	Указатель базы
	<b>SI</b>	Индекс источника
	<b>DI</b>	Индекс назначения

Имеется несколько способов задания операнда, находящегося в оперативной памяти, они называются **режимами адресации**.

Способы адресации микропроцессора 8x86: регистровый, непосредственный, прямой, косвенный, базовый, прямой с индексированием, базовый с индексированием.

(<http://studopedya.ru/1-40528.html> – Режимы адресации 16-разрядного микропроцессора Intel-8086).

В данном пособии рассматриваются следующие способы адресации:

- **регистровая адресация** – операнд находится в одном из регистров процессора, а в команде указан этот регистр;
- **непосредственная** – в команде указывается константа; значение константы в качестве операнда-источника;
- **косвенная адресация** – в команде указывается регистр, содержащий адрес аргумента;
- **базовая (относительная адресация)** – адрес аргумента равен сумме содержимого регистра и константы, задающей смещение.

### *Команды процессора*

В общем случае система команд процессора включает в себя следующие четыре основные группы команд:

1. **Команды пересылки данных;**
2. **Арифметические команды ;**
3. **Команды переходов.**
4. **Логические команды ;**

## **1. Команды пересылки данных**

### **1.1. mov - пересылка данных**

Общий вид: **mov** <операнд1>,< операнд2>

**mov** приёмник, источник

Копирует содержимое источника в приёмник, источник не изменяется, работает как с однобайтовыми, так и с двухбайтовыми словами.

Операнды команды **mov** могут быть как регистрами, так и переменными (одновременно оба операнда не могут быть переменными).

**Пример 1.** Команда пересылки

Исходное состояние регистров: **AX=0000; BX=0000.**

**mov AX, 5** присваивает регистру **AX** значение 5. Непосредственный способ адресации.

**mov BX, AX** присваивает регистру **BX** значение, хранящееся в регистре **AX**. Регистровый способ адресации.

Мнемоника команды	Содержимое AX после выполнения команды	Содержимое BX после выполнения команды
<b>mov AX, 5</b>	0005	0000
<b>mov BX, 6</b>	0005	0006
<b>mov BX, AX</b>	<b>0005</b>	<b>0005</b>

**1.2. XCHG** меняет местами содержимое двух операндов. Работает с регистрами, подрегистрами.

Общий вид: **xchg** <операнд1>,< операнд2>

**Пример 2.** Работа **XCHG** с подрегистрами.

	До выполнения команды обмена al=2b, ah=1a:	После выполнения команды обмена <b>xchg al, ah</b>																		
<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="text-align: center;">AH</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">AX</td> <td style="text-align: center;">AL</td> </tr> <tr> <td style="text-align: center;">15</td> <td style="border-left: 1px solid black; border-right: 1px solid black; text-align: center;">8 7</td> <td style="text-align: center;">0</td> </tr> </table>	AH	AX	AL	15	8 7	0	<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="text-align: center;">1a</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">2b</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">15</td> <td style="border-left: 1px solid black; border-right: 1px solid black; text-align: center;">8 7</td> <td style="text-align: center;">0</td> </tr> </table>	1a	2b	0	15	8 7	0	<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="text-align: center;">2b</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px;">1a</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">15</td> <td style="border-left: 1px solid black; border-right: 1px solid black; text-align: center;">8 7</td> <td style="text-align: center;">0</td> </tr> </table>	2b	1a	0	15	8 7	0
AH	AX	AL																		
15	8 7	0																		
1a	2b	0																		
15	8 7	0																		
2b	1a	0																		
15	8 7	0																		

## 2. Арифметические команды

Арифметические команды выполняют операции сложения, вычитания, умножения, деления, увеличения на единицу (инкрементирования), уменьшения на единицу (декрементирования) и т.д.

**2.1. Сложение: add**

Общий вид: **add** <операнд1>,< операнд2>

**add** – выполняет сложение *операнда 1* и *операнда 2*, результат заносится в *операнда 1* (*операнда 1*= *операнда 1*– *операнда 2* ).

*Операнда 2* не изменяется. Изменяются флаги. В качестве *операнда 1* может быть регистр или ячейка памяти; в качестве *операнда 2* – регистр, ячейка памяти, константа.

### Пример 3. Команда сложения

Мнемоника команды	Содержимое AX после выполнения команды	Содержимое BX после выполнения команды
	0000	0000
<b>add BX, 5</b>	0000	0005
<b>add AX, 7</b>	0007	0005
<b>add BX, AX</b>	0007	<b>0012</b>
<b>add AX, BX</b>	<b>0019</b>	0012

#### 2.2. Вычитание: *sub*

Общий вид: **sub** <операнд1>, <операнд2>

**sub** – выполняет вычитание: операнда 1 = операнда 1 – операнда 2 .

*Операнда 2* не изменяется. Изменяются флаги. В качестве *операнда 1* может быть регистр или ячейка памяти; в качестве *операнда 2* – регистр, ячейка памяти, константа.

#### Пример 4. Команда вычитания.

Исходное состояние регистров: AX=0000; BX=0005; CX=0000; DX=0008;

Мнемоника команды	Содержимое DX после выполнения команды	Содержимое BX после выполнения команды
<b>sub DX, 2</b>	<b>0006</b>	0005
<b>sub DX, BX</b>	<b>0001</b>	0005
<b>sub DX, 3</b>	<b>FFFE</b>	0005

**FFFE** - отрицательный результат, происходит изменение флагов.

#### 2.3. Умножение: *mul*

Общий вид: **mul** <операнд>

Производит беззнаковое умножение содержимого аккумулятора на указанный операнд.

При *однобайтовом (байтовое)* умножении: умножается содержимое AL на операнд, результат - в AX.

#### Пример 4. Умножение однобайтовое.

Исходное состояние регистров: AX=0000; BX=0000; CX=0000; DX=0000;

Мнемоника команды	Содержимое AX после выполнения команды	Содержимое BX после выполнения команды
<b>mov AX, 3</b>	0003	0000
<b>mov BX, 2</b>	0003	0002
<b>mul BX</b>	<b>0006</b>	0002



При *двубайтовом (слово)* умножении: умножается содержимое AX на операнд, результат (двойное слово) размещается в двух регистрах: в DX старшие разряды результата, в AX.- младшие разряды результата.

**Пример 5.** Умножение двубайтовое.

Исходное состояние регистров: AX=0000; BX=0000; CX=0000; DX=0000;

Мнемоника команды	Содержимое AX после выполнения команды	Содержимое BX после выполнения команды	Содержимое DX после выполнения команды
<b>mov AX,1111</b>	1111	0000	0000
<b>mov BX,1111</b>	1111	1111	0000
<b>mul BX</b>	<b>4321</b>	1111	<b>0123</b>

$$1111_{16} \cdot 1111_{16} = 123\ 4321_{16}$$

**2.4. Деление: div**

Общий вид: **div** <операнд>

Осуществляет деление содержимого аккумулятора на указанный операнд (без учета знаков).

Операнд может быть задан в регистре общего назначения или ячейке памяти.

При *байтовом* делении: делится содержимое AX на операнд; результат - целая часть деления размещается в регистре AL, а остаток деления – в регистр AH.

**Пример 6.** Деление однобайтовое.

Исходное состояние регистров: AX=0000; BX=0000; CX=0000; DX=0000;

Мнемоника команды	Содержимое AX после выполнения команды		Содержимое BX после выполнения команды
	AH	AL	
<b>mov AX,9</b>	00	09	0000
<b>mov BX,2</b>	00	09	0002
<b>div BX</b>	<b>01</b>	<b>04</b>	0002

При *двухбайтовом (слово)* делении: делится содержимое регистров DX:AX на операнд. Результат размещается: целая часть деления – в регистре AX, остаток деления - в регистре DX.

### Пример 7. Деление двухбайтовое.

Найти результат деления числа 00ED 4389(H) на число 6B1(H).

#### Решение:

Число 00ED4389(H) размещается в следующих регистрах: DX= 00ED; AX= 4389.

00ED 4389(H) / 6B1(H) = целая часть: 2375(H); остаток: 1A4(H)

Исходное состояние регистров: AX=0000; BX=0000; CX=0000; DX=0000;

Мнемоника команды	Содержимое AX после выполнения команды	Содержимое BX после выполнения команды	Содержимое DX после выполнения команды
<b>mov AX, 4389</b>	4389	0000	0000
<b>mov DX, 00ED</b>	4389	0000	00ED
<b>mov BX, 6B1</b>	4389	06B1	00ED
<b>div BX</b>	<b>2375</b>	06B1	<b>01A4</b>

### 2.5. Инкремент: *inc*

Общий вид: **inc** <операнд>

Увеличивает содержимое операнда на 1. Операнд может быть задан в регистре общего назначения или в ячейке памяти. Команда работает как с однобайтовыми, так и двухбайтовыми операндами, без учета знака.

#### Пример 8. Инкремент с однобайтовыми операндами.

Исходное состояние регистров:  
AX=0000; BX=0000; CX=0000; DX=0000;

Мнемоника команды	Содержимое AX после выполнения команды	
	AH	AL
<b>mov AX,5</b>	00	05
<b>inc AL</b>	00	<b>06</b>
<b>inc AH</b>	<b>01</b>	06

#### Пример 9. Инкремент с двухбайтовыми операндами.

Исходное состояние регистров:  
AX=0000; BX=0000; CX=0000; DX=0000;

Мнемоника команды	Содержимое BX после выполнения команды
<b>mov BX, 8</b>	0008
<b>inc BX</b>	0009
<b>inc BX</b>	000A

### 2.6. Декремент: *dec*

Общий вид: **dec** <операнд>

Уменьшает содержимое операнда на 1. Операнд может быть задан в регистре общего назначения или в ячейке памяти. Команда работает как с однобайтовыми, так и двухбайтовыми операндами, без учета знака.

**Пример 10.** Декремент с однобайтовыми операндами.

Исходное состояние регистров:  
 AX=0000; BX=0000; CX=0000; DX=0000;

Мнемоника команды	Содержимое CX после выполнения команды	
	CH	CL
<b>mov CX, 0907</b>	09	07
<b>dec CL</b>	09	<b>06</b>
<b>dec CH</b>	<b>08</b>	06

**Пример 11.** Декремент с двухбайтовыми операндами.

Исходное состояние регистров:  
 AX=0000; BX=0000; CX=0000; DX=0000;

Мнемоника команды	Содержимое CX после выполнения команды
<b>mov CX, 7</b>	0007
<b>dec CX</b>	0006
<b>dec CX</b>	0005

### 3. Команды переходов

Команды переходов предназначены для изменения обычного порядка последовательного выполнения команд. С их помощью организуются переходы к подпрограммам и возврат из них, реализация ветвлений и циклов в программах. Команды переходов всегда меняют содержимое счетчика команд. Переходы могут быть условными и безусловными. Именно эти команды позволяют строить сложные алгоритмы обработки информации.

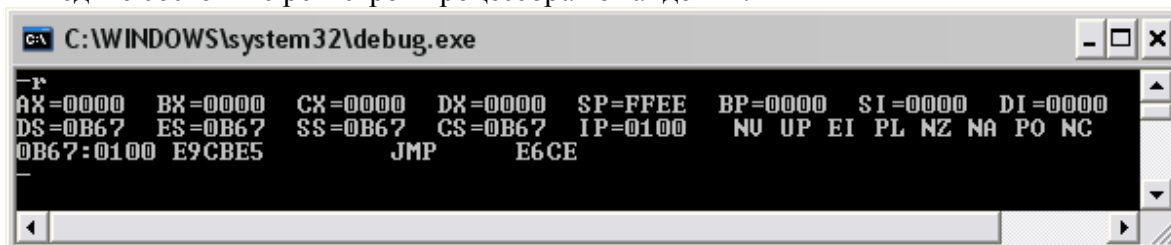
### 4. Логические команды

Логические команды производят над операндами логические операции, например, логическое И, логическое ИЛИ, исключающее ИЛИ, очистку, инверсию, разнообразные сдвиги (вправо, влево, арифметический сдвиг, циклический сдвиг). Этим командам, как и арифметическим, требуется один или два входных операнда, и формируют они один выходной операнд команды.

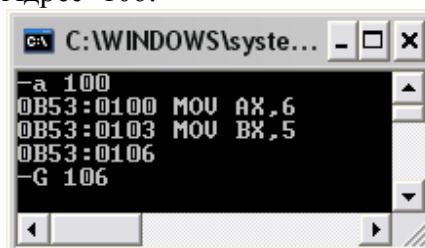
### Задание № 1

Запустите Непосредственный регистровый способы адресации.

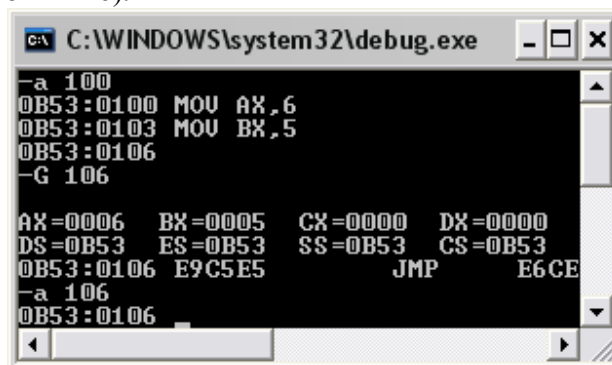
1. Запустите программу **debug.exe**.
2. Выведите состояние регистров процессора командой **R**:



3. Введите, начиная с адреса 100 в ОП программу, которая выполнит следующие действия:
  - a. загрузит в регистры *AX* и *CX* числа *1(H)* и *10(H)* соответственно
  - b. перешлет в регистр *BX* содержимое регистра *AX*
4. Запустите программу на выполнение командой **G <адрес>**, где **адрес** – это адрес первого командного байта той команды, которая следует за вашей программой. Например, для программы, приведенной ниже, Адрес=106.



5. Обратите внимание на изменения состояний регистров общего назначения .
6. Добавьте в программу команду, которая перешлет в регистр *DX* содержимое регистра *CX*.  
Для этого:
  - a. перейдите по адресу последней свободной команды в программа, с помощью команды **A<Адрес>**. В данном примере – это адрес 106.(см. рисунок ниже).



- b. Наберите нужную команду.
7. Выведите на экран команды программы. Для этого воспользуйтесь командой **U <адрес начала, адрес конца>**, где *адрес нач* – адрес первого байта программы, *адрес кон* – адрес первого байта последней команды программы.

Например, для программы, представленной в рассматриваемом примере, вывод на экран команд программы: **U 100, 103**

**-U 100, 103**

**0B53:0100 B80600 MOV AX,0006**

**0B53:0103 BB0500 MOV BX,0005**

8. Рассчитайте длину программы в байтах и количество тактов процессора для выполнения программы. Для этого:

- а. Подготовьте и заполните следующую таблицу для разрабатываемой программы. Например, для данного примера таблица выглядит следующим образом (см. Таблица 1):

Таблица 1

Адрес	Код	Мнемоника	Длина команды ( байт)	Время выполнения в машинных тактах (Кол-во тактов)
<b>0B53:0100</b>	<b>B80600</b>	<b>MOV AX,0006</b>	3	4
<b>0B53:0103</b>	<b>BB0500</b>	<b>MOV BX,0005</b>	3	4
<b>Итого для программы</b>			6	8

Данные для полей «Длина команды, байт» и «Кол-во тактов» представлены в Приложении 1 данного раздела.

**Задание №2.** Разработайте программу, используя регистровый способ адресации для сложения содержимого двух регистров.

№ варианта	Размещение данных в регистрах	№ варианта	Размещение данных в регистрах
<b>1</b>	Исходные данные: AX = 9, BX = A1; результат в регистре BX	<b>8</b>	Исходные данные: BX = 100, CX = AA; результат в регистре CX
<b>2</b>	Исходные данные: BX = 1C, CX = 29; результат в регистре CX	<b>9</b>	Исходные данные: AX = 109, BX = 31; результат в регистре AX
<b>3</b>	Исходные данные: DX = 2A, CX = 3; результат в регистре DX	<b>10</b>	Исходные данные: CX = 65, DX = 4D1; результат в регистре DX
<b>4</b>	Исходные данные: BX = 51, AX = 231; результат в регистре AX	<b>11</b>	Исходные данные: AX = 76, BX = 34; результат в регистре AX
<b>5</b>	Исходные данные: AX = 4D1, CX = 2; результат в регистре AX	<b>12</b>	Исходные данные: CX = 111, AX = 7A1; результат в регистре CX
<b>6</b>	Исходные данные: BX = 3, CX = 7B7; результат в регистре BX	<b>13</b>	Исходные данные: DX = 72, BX = A4; результат в регистре DX
<b>7</b>	Исходные данные: DX = B5, AX = 71; результат в регистре DX	<b>14</b>	Исходные данные: CX = 12, DX = B58; результат в регистре CX

Выведите программу на экран. Вычислите *длину* программы и *время* ее исполнения в машинных тактах.

**Задание №1.**

1.1. Разработайте программу для выполнения указанных действий с регистрами общего назначения:

№ варианта	Действия с регистрами процессора	№ варианта	Действия с регистрами процессора
<b>1</b>	<ul style="list-style-type: none"> <li>○ в регистр АХ загружается число 200, в регистр СХ – число 51</li> <li>○ <i>вычитается</i> из содержимого регистра АХ содержимое регистра СХ</li> <li>○ увеличивается результат на единицу</li> <li>○ <i>делится</i> содержимое регистра АХ на содержимое регистра СХ</li> </ul>	<b>8</b>	<ul style="list-style-type: none"> <li>○ в регистр АХ загружается число ВА1, в регистр ВХ – число 67</li> <li>○ <i>вычитается</i> из содержимого регистра АХ содержимое регистра ВХ</li> <li>○ <i>делится</i> содержимое регистра АХ на содержимое регистра ВХ</li> <li>○ уменьшается результат на единицу</li> </ul>
<b>2</b>	<ul style="list-style-type: none"> <li>○ в регистр DX загружается число 89А, в регистр СХ – число 12</li> <li>○ <i>делится</i> содержимое регистра DX на содержимое регистра СХ</li> <li>○ <i>вычитается</i> из содержимого регистра результат деления</li> </ul>	<b>9</b>	<ul style="list-style-type: none"> <li>○ в регистр DX загружается число АЗВ, в регистр СХ – число АВС</li> <li>○ <i>вычитается</i> из содержимого регистра СХ содержимое регистра DX</li> <li>○ увеличивает результат на единицу</li> <li>○ <i>делится</i> содержимое регистра СХ на содержимое регистра DX</li> </ul>
<b>3</b>	<ul style="list-style-type: none"> <li>○ в регистр АХ загружается число 199, в регистр ВХ – число 5В</li> <li>○ <i>делится</i> содержимое регистра АХ на содержимое регистра ВХ</li> <li>○ <i>вычитается</i> из содержимого регистра ВХ содержимое регистра АХ</li> <li>○ увеличивается результат на единицу</li> </ul>	<b>10</b>	<ul style="list-style-type: none"> <li>○ в регистр ВХ загружается число ВА, в регистр СХ – число 70</li> <li>○ <i>вычитается</i> из содержимого регистра ВХ содержимое регистра СХ</li> <li>○ <i>делится</i> содержимое регистра ВХ на содержимое регистра СХ</li> </ul>
<b>4</b>	<ul style="list-style-type: none"> <li>○ в регистр ВХ загружается число 23, в регистр DX – число 4С2</li> <li>○ <i>делится</i> содержимое регистра DX на содержимое регистра ВХ</li> <li>○ <i>вычитается</i> из содержимого регистра DX результат деления</li> </ul>	<b>11</b>	<ul style="list-style-type: none"> <li>○ в регистр DX загружается число 6Е2, в регистр АХ – число 5Е3</li> <li>○ <i>вычитается</i> из содержимого регистра DX содержимое регистра АХ</li> <li>○ увеличивает результат на единицу</li> <li>○ <i>делится</i> содержимое регистра АХ на содержимое регистра DX</li> </ul>
<b>5</b>	<ul style="list-style-type: none"> <li>○ в регистр DX загружается число 34, в регистр АХ – число 98А</li> <li>○ <i>делится</i> содержимое регистра АХ на содержимое регистра DX</li> <li>○ <i>вычитается</i> из содержимого регистра DX содержимое регистра АХ</li> <li>○ увеличивает результат на единицу</li> </ul>	<b>12</b>	<ul style="list-style-type: none"> <li>○ в регистр АХ загружается число 3А, в регистр СХ – число 213</li> <li>○ <i>вычитается</i> из содержимого регистра АХ содержимое регистра СХ</li> <li>○ уменьшается результат на единицу</li> <li>○ <i>делится</i> содержимое регистра АХ на содержимое регистра СХ</li> </ul>

№ варианта	Действия с регистрами процессора	№ варианта	Действия с регистрами процессора
6	<ul style="list-style-type: none"> <li>○ в регистр ВХ загружается число 54, в регистр СХ – число 2Е5</li> <li>○ <i>вычитается</i> из содержимого регистра СХ содержимое регистра ВХ</li> <li>○ <i>делится</i> содержимое регистра СХ на содержимое регистра ВХ</li> </ul>	13	<ul style="list-style-type: none"> <li>○ в регистр АХ загружается число 5С3, в регистр ВХ – число 232</li> <li>○ <i>делится</i> содержимое регистра АХ на содержимое регистра ВХ</li> <li>○ уменьшается результат на единицу</li> <li>○ <i>вычитается</i> из содержимого регистра ВХ содержимое регистра АХ</li> </ul>
7	<ul style="list-style-type: none"> <li>○ в регистр ДХ загружается число Е1, в регистр СХ – число 15</li> <li>○ <i>вычитается</i> из содержимого регистра ДХ содержимое регистра СХ</li> <li>○ <i>делится</i> содержимое регистра ДХ на содержимое регистра СХ</li> </ul>	14	<ul style="list-style-type: none"> <li>○ в регистр ВХ загружается число С1С, в регистр СХ – число АА</li> <li>○ <i>делится</i> содержимое регистра ВХ на содержимое регистра СХ</li> <li>○ <i>вычитается</i> из содержимого регистра ВХ результат деления</li> </ul>

1.2. Вычислите длину вашей программы и время ее исполнения в машинных тактах.

## Задание №2.

1.1. Разработайте программу для вычисления значения выражений, согласно Вашему варианту, используя регистры общего назначения. Регистры для хранения исходных данных и результата определите самостоятельно. Исходные данные представлены в десятичной системе счисления.

Вариант	Выражение	Вариант	Выражение
1	$(7812 * 7726 + 25) - 1$	8	$96 * 98122 + 23 * 16 - 1$
2	$7412 * 124 + 6523 * 565 + 1$	9	$9999 + (3659 * 6315 + 1)$
3	$(5612 * 814 - 393) * 48 + 1$	10	$(15248 + 2635) * 18 - 1$
4	$4332 * 57811 - 23 * 12 + 1$	11	$1299 * 4521 - (523 * 326 - 1)$
5	$(1712 * 19999 - 1) + 45781$	12	$(58889 - 1253) * 111 + 1$
6	$1334 * 1421 + 2643 * 23 + 1$	13	$(2993 * 3354 + 89545) + 1$
7	$8989 + (7894 * 5111 - 1)$	14	$4813 * 214 + 1653 * 834 - 1$

1.2. Вычислите длину вашей программы и время ее исполнения в секундах при тактовой частоте 2,33 ГГц.

## КОСВЕННЫЙ СПОСОБ АДРЕСАЦИИ

**Косвенная адресация** – в команде указывается регистр, содержащий адрес аргумента.

Косвенные регистровые операнды нужно заключать в квадратные скобки, чтобы отличить их от регистровых операндов.

**Пример 1.** Загрузка в регистре общего назначения содержимого ячейки памяти.

**Дано:**

1. Шестнадцатеричные числа 15H, 10H, 5A5AH размещены в оперативной памяти, начиная с ячейки 90:

```
-A90
XXXX:0090 dw 15
XXXX:0092 dw 10
XXXX:0094 dw 5A5A
XXXX:0096
```

dw - определить слово

2. Исходное состояние регистров:

```
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B78 ES=0B78 SS=0B78 CS=0B78 IP=0100
```

**Необходимо:** используя косвенный способ адресации, загрузить содержимое ячейки оперативной памяти 0090H в регистр BX.

**Решение:**

```
-A100
0100 MOV BX,[90]
0104
```

**[90]** - обращение к ячейке 0090 оперативной памяти, используя косвенный способ адресации. После запуска программы на выполнение с адреса 104 содержимое регистров изменилось:

```
AX=0000 BX=0015 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B78 ES=0B78 SS=0B78 CS=0B78 IP=0104
```

**Пример 2.** Размещение содержимого регистра общего назначения в ячейке памяти.

Используя данные примера №1, записать в ячейку памяти 0096 содержимое регистра BX:

```
0104 MOV [96],BX
```





**Пример 3.** Сложение содержимого регистра и содержимого ячейки памяти, используя косвенный способ адресации.

*Дано:*

Состояние регистров данных процессора после примера №2:

AX=0000 **BX=0015** CX=0000 DX=0000

**Необходимо:** используя косвенный способ адресации, сложить содержимое ячейки оперативной памяти 0094H с содержимым регистра BX. Результат сложения - в BX.

*Решение:*

ADD BX,[94]

После выполнения команды сложения:  $BX = BX + [94] = 0015 + 5A5A = 5A6F$ , содержимое регистра BX = 5A6F

*Косвенный способ адресации*

**Задание №1.** Составьте программу, которая копирует одномерный массив из шести элементов (предварительно размещенный в ячейках памяти начиная с 90), в ячейки оперативной памяти, начиная с адреса 300 Н. Используйте *косвенный* способ адресации.

**Задание №2.** Для программы из задания №1 вычислите длину программы в байтах и время ее выполнения в тактах, заполнив следующую таблицу:

Таблица 1

Адрес	Код команды	Мнемоника	Длина команды (байт)	Время выполнения в машинных тактах (количество тактов)
<i>Итого для программы</i>				

**Пример расчета длины команды с использованием косвенной адресации:**

Например, команда MOV AX, [90] («переслать из ячейки памяти 90 в регистр AX»). Следовательно, в таблице [Приложения](#) необходимо отыскать **описание** Память – регистр, и соответствующее **Число тактов: 8+EA**).

**EA** – это эффективный адрес. Время вычисления EA (в тактах) определяется значениями полей **R/M** и **MOD** кода операции команды.

**1. Определите значение полей RM и MOD.** Для этого:

**1.1.** Выведите на экран команды программы. Для этого воспользуйтесь командой **U <адрес начала, адрес конца>**, где *адрес нач* – адрес первого байта программы, *адрес кон* – адрес первого байта последней команды программы.

**1.2.** Команда MOV AX [90] (ее код A19000) имеет внутреннее представление:

$$A19000 = \underbrace{1010\ 0001}_{1\text{-й байт}} \underbrace{1001\ 0000}_{2\text{-й байт}} \underbrace{0000\ 0000}_{3\text{-й байт}} B$$

где:

*1-й байт:* 1010 0001 – код операции:

d = 0 – направление передачи из микропроцессора в оперативную память,

w = 1 – команда оперирует словом;

*2-й байт:* 1001 0000

10 – значение поля **MOD**, команда содержит два байта смещения;

01 0 – адрес регистра, регистр AX;

000 – значение поля **R/M**,

*3-й байт disp:* 0000 0000 – значение смещения.

1.3. Из табл. 2 для колонки «Значение поля **R/M**» определяем строку со значением **000**. Соответственно для колонки «**MOD =10**», в первой строке находим время вычисления **EA(в тактах)**, следовательно,  $EA = (BX)+(SI)+DISP16$  и на его вычисление процессору потребуется **11** тактов.

Таблица 2

Значение поля <b>R/M</b>	<b>MOD = 00</b>	Время вычисления <b>EA (в тактах)</b>	<b>MOD = 01</b>	Время вычисления <b>EA (в тактах)</b>	<b>MOD =10</b>	Время вычисления <b>EA (в тактах)</b>
<b>000</b>	$(BX)+(SI)$	7	$(BX)+(SI)+DISP8$	11	$(BX)+(SI)+DISP16$	<b>11</b>
001	$(BX)+(DI)$	8	$(BX)+(DI)+DISP8$	12	$(BX)+(DI)+DISP16$	12
010	$(BP)+(SI)$	8	$(BP)+(SI)+DISP8$	12	$(BP)+(SI)+DISP16$	12
011	$(BP)+(DI)$	7	$(BP)+(DI)+DISP8$	11	$(BP)+(DI)+DISP16$	11
100	$(SI)$	5	$(SI)+DISP8$	9	$(SI)+DISP16$	9
101	$(DI)$	5	$(DI)+DISP8$	9	$(DI)+DISP16$	9
110	$DISP16$	6	$(BP)+DISP8$	9	$(BP)+DISP16$	9
111	$(BX)$	5	$(BX)+DISP8$	9	$(BX)+DISP16$	9

Таким образом, команду **MOV AX [90]** процессор выполнит за  $8+11 = 19$  машинных тактов, причем большую часть из них занимает вычисление эффективного адреса.

Аналогично для других команд программы.

*Базовый способ адресации*

**Задание №1**

**1.1.** Изучите материал по *базовому способу* адресации данных.

**Относительная адресация (базовая)** – адрес аргумента равен сумме содержимого регистра и константы, задающей смещение.

Основное применение базового способа адресации – доступ к элементам структур данных. Содержимое **базового регистра** указывает на **начало структуры данных**, а требуемый элемент адресуется с помощью смещения (расстояния от базы), содержащегося в команде. В этой работе в качестве базового для адресации данных используется регистр ВХ.

**Пример 1**

*Дано:*

1. Шестнадцатеричные числа 01Н, 10Н, А5А5Н, ВВВВН, 77ССН размещены в оперативной памяти, начиная с ячейки 90:

```
-А90
    0090 dw 01
    0092 dw 10
    0094 dw А5А5
    0096 dw ВВВВ
    0098 dw 77СС
```

2. Исходное состояние регистров:

```
АХ=0000 ВХ=0000 СХ=0000 ДХ=0000 SP=FFEE ВР=0000 SI=0000 DI=0000
DS=0В78 ES=0В78 SS=0В78 CS=0В78 IP=0100
```

**Необходимо:** используя базовый способ адресации, загрузить содержимое ячейки оперативной памяти 0090Н в регистр ВХ, содержимое ячейки оперативной памяти 0092Н в регистр СХ.

**Решение:**

1. Загрузить в регистр ВХ значение базы (адрес ячейки - 90):

```
-А100
0100 MOV ВХ,90
0103
```

2. Загрузить содержимое ячейки 0090Н оперативной памяти в регистр АХ:

```
0103 MOV АХ,[ВХ]
0105
```

3. Загрузить содержимое ячейки 0092Н оперативной памяти в регистр СХ.

```
-А106
0105 MOV СХ,[ВХ+2]
0108
```

*Итог:*

Данные:

0090 dw 01  
0092 dw 10  
0094 dw A5A5  
0096 dw BBBB  
0098 dw 77CC

Программа:

0100 MOV BX,90  
0103 MOV AX,[BX]  
0105 MOV CX,[BX+2]

Регистры процессора после выполнения программы:

**AX=0001 BX=0090 CX=0010 DX=0000 BP=0000**

**1.2.** Добавьте к полученной программе команды, с помощью которых в регистр AX будет записано содержимое ячейки оперативной памяти 0094, в регистр CX - содержимое ячейки оперативной памяти 0098, в регистр DX - содержимое ячейки оперативной памяти 0096.

**1.3.** Рассчитайте длину программы в байтах и время выполнения в машинных тактах. Используйте табличные данные [Приложения 1](#).

**Задание №2.** Рассматривая заданную последовательность шестнадцатеричных чисел 117(H), 8(H), 1B3C(H), 23D1(H), 5EFF(H) как одномерный массив, разработайте программу с базовым способом адресации данных, копирующую массив в ячейки оперативной памяти, начиная с адреса 200H текущего сегмента.

### **Задание №3**

**3.1.** Разработайте программу, увеличивающую на 5(H) каждый элемент массива, хранящегося в оперативной памяти, начиная с адреса 200(H) текущего сегмента. Результат работы программы – массив, хранящийся в ячейках оперативной памяти, начиная с адреса 300(H) текущего сегмента. Используйте базовый способ адресации данных.

**3.2.** Вычислите время выполнения программы *в секундах* при тактовой частоте процессора 2,33 ГГц.

**3.3.** Сколько байтов оперативной памяти требуется для работы программы?

### ОРГАНИЗАЦИЯ ВЕТВЛЕНИЙ В ПРОГРАММАХ

Переход - это передача управления другой команде. Переходы разделяют на: *безусловные и условные*.

#### I. Безусловные переходы

Команда **JMP**.

Общий вид: **JMP** <адрес>

Операндом может быть: непосредственно значение в памяти; регистр, содержащий адрес; непосредственно значение адреса.

**Пример 1.** Использование команды безусловного перехода **JMP** (переход осуществляется к команде, расположенной после команды **JMP**).

```
0B78:0100    MOV    AX, 003D
0B78:0103    MOV    BX, 0004
0B78:0106    JMP  010B
0B78:0108    MOV    CX, 0057
0B78:010B    MOV    DX, 0111
0B78:010E
```

Командой **JMP 010B** осуществляется переход к команде **MOV DX, 0111** по адресу **010B**.

Состояние регистров после выполнения программы:

```
AX=003D BX=0004 CX=0000 DX=0111 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B78 ES=0B78 SS=0B78 CS=0B78 IP=010E
```

**Пример 2.** Использование команды безусловного перехода **JMP** (переход осуществляется к команде, расположенной до команды **JMP**).

```
0B78:0100    MOV    AX, 003D
0B78:0103    MOV    BX, 0004
0B78:0106    JMP  0103
0B78:0108    MOV    CX, 0057
0B78:010B    MOV    DX, 0111
0B78:010E
```

Командой **JMP 0103** осуществляется переход к команде **MOV BX, 0004** по адресу **0103**. В данном примере программа свою работу завершить корректно не сможет - будут «до бесконечности» выполняться команды: *0103 MOV BX, 0004* и *0106 JMP 0103*.

#### II. Условные переходы

Условная команда - это команда, выполнение которой зависит от каких-либо условий. Например, требуется определенная установка флажков или необходимо наличие нуля в некотором регистре. Самыми употребительными из всех условных команд являются команды условной передачи управления.

Для реализации условного перехода используют две инструкции:

1. Команда Сравнения **СМР**;
2. Команда Условного перехода (**JA, JB, JZ**, .. и т.п. – см. Приложение 2).

1. Команда Сравнения **СМР**  
**СМР <операнд1>, <операнд2>**.

В качестве первого операнда может выступать *регистр* или *значение памяти* любого размера (1, 2, 4, 8 байт). В качестве второго операнда может выступать *регистр*, *значение памяти* или *непосредственное значение*.

**Пример 3.** Некоторые форматы команды сравнения **СМР**:

**СМР AX, BX** – сравнение содержимого регистров;

**СМР AX, 6** – сравнение содержимого регистра **AX** с непосредственным значением = 6;

**СМР [90], AX** – сравнение содержимого ячейки памяти по адресу 90 с содержимым регистра **AX**.

2. Команды условного перехода (**JA, JB, JZ**, .. и т.п. – см. Приложение 2):

**JA<операнд>, JB<операнд>, JZ<операнд>, и т.п.**

В качестве операнда - адрес в программе, куда требуется осуществить переход.

Если условие *сравнения* «истинно», то **JA** (или другая команда условного перехода) осуществляет переход по указанному адресу (состояние флагов не изменяется - Приложение 3).

Если условие *сравнения* «ложь», то **JA** (или другая команда условного перехода) осуществляет переход к следующей команде программы (состояние флагов изменяется).

**Пример 4.** Использование команды сравнения и команды условного перехода.

*Исходное состояние регистров:*

**AX=0028 BX=0027**

*Необходимо:* Записать в регистр общего назначения **DX** число 2, если **AX > BX**, иначе - записать в **CX** число 1.

**Решение:**

Программа:

**0100 СМР AX, BX** – сравнение;

**0102 JA 0107** – команда условного перехода; если **AX > BX**, то передача управления команде по адресу 107, иначе переход к следующей команде. Команда **JA** занимает 2 байта и выполняется процессором за 16/4 тактов (см. Приложение 1). 16 тактов на переход к команде, если условие сравнения истинно, и 4 такта - в противном случае.

**0104 MOV CX, 0001**

**0107 MOV DX, 0002**

**010A**

*Состояние регистров после выполнения программы:*

**AX=0025 BX=0027 CX=0000 DX=0002 SP=FFEE BP=0000 SI=0000 DI=0000**  
**DS=0B72 ES=0B72 SS=0B72 CS=0B72 IP=010A NV UP EI NG NZ AC PO CY**  
**0B72:010A 5F POP DI**

**ВЕТВЛЕНИЯ**

**Задание № 1**

**1.1.** Изучите программы 1 и 2.

Программа 1	Программа 2
0100 MOV AX,05	0100 MOV AX,05
0103 CMP AX,05	0103 CMP AX,07
0106 JZ 10D	0106 JZ 10D
0108 MOV BX,77	0108 MOV BX,77
010B JMP 110	010B JMP 110
010D MOV BX,88	010D MOV BX,88
0110 MOV AX,99	0110 MOV AX,99
0113	0113

Ответьте на следующие вопросы:

1. Какие значения сравниваются в программе 1 командами по адресу 103 и 106?
2. Какие значения сравниваются в программе 2 командами по адресу 103 и 106?
3. Какую задачу решают представленные программы?

**1.2.** Рассчитайте время выполнения программ 1 и 2 в машинных тактах, заполнив следующую таблицу для каждой из программ:

Адрес	Мнемоника	Значения		Время выполнения программы №1 в машинных тактах
		Регистра AX	Регистра BX	

**Задание № 2**

Разработайте программу, которая сравнивает числа, находящиеся в регистрах BX и CX, и записывает большее из них в регистр DX, а меньшее из них – в регистр AX.

**Задание № 3**

Разработайте программу, которая сравнивает числа, находящиеся в регистрах BH, BL и CH и записывает в регистр DX меньшее из них.



Организация ветвлений

**Задание № 1.** Найдите значение выражения, используя команды условного перехода.

Вариант	Выражение	Вариант	Выражение
1	$K = \begin{cases} X + 12, & X > 15 \\ Y * 32 + 4, & X \leq 15 \end{cases}$	8	$K = \begin{cases} 7, & Y \leq 78 \\ Y / 2 - 10, & Y > 78 \end{cases}$
2	$K = \begin{cases} X / 5, & X \Rightarrow 5 \\ Y + 17, & X < 5 \end{cases}$	9	$K = \begin{cases} X + 21, & Y < 21 \\ Y * 2 - 42, & Y \geq 21 \end{cases}$
3	$K = \begin{cases} X - 1, & X \leq 7 \\ Y * 18 + 4, & X > 7 \end{cases}$	10	$K = \begin{cases} X * 13 - 2, & X > 9 \\ Y + 44, & X \leq 9 \end{cases}$
4	$K = \begin{cases} 6 * X - 3, & X > 9 \\ Y + 2, & X \leq 9 \end{cases}$	11	$K = \begin{cases} X / 5 + 5, & X > 3 \\ Y, & X \leq 3 \end{cases}$
5	$K = \begin{cases} X + 4, & X \Rightarrow 4 \\ Y * 2 + 4, & X < 4 \end{cases}$	12	$K = \begin{cases} X / 7 - 15, & Y \leq 68 \\ 45, & Y > 68 \end{cases}$
6	$K = \begin{cases} X, & Y < 2 \\ Y / 2 + 1, & Y \geq 2 \end{cases}$	13	$K = \begin{cases} X + 32, & X \geq 114 \\ Y * 2 - 1, & X < 114 \end{cases}$
7	$K = \begin{cases} 2 * X + 3, & X > 8 \\ Y - 12, & X \leq 8 \end{cases}$	14	$K = \begin{cases} X - 8, & X > 9 \\ Y * 3 + 10, & X \leq 9 \end{cases}$

**Задание № 2**

**2.1. Изучите материал по организации циклов в программах.**

Организация циклов

**1. Изучите программу, представленную ниже (программа №1):**

```

-A 100
0100 MOV AX,01          4 такта
0103 MOV CX,10         4 такта
0106 INC AX            2 такта
0107 DEC CX            2 такта
0108 JNZ 106           16/4 такта
010A
    
```

- После запуска программы (-G10A) получен следующий результат:  
**AX=0011 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000**  
**DS=XXXX ES=XXXX SS=XXXX CS=XXXX IP=0113 NU UP EI PL NZ NA**  
**PE NC**

- Обратите внимание на команды, выделенные зеленым маркером. Они повторяются 16 раз, так как значение регистра CX=10(H) до начала «повтора» команд. После того как значение CX стало равно нулю, флаговый регистр ZF (флаг нуля, см. Приложение 3) изменил свое состояние с ZR на **NZ**.
- Командой U 100,10A выведена на экран программа с адреса 100 до адреса 10A:

```

0100 B80100      MOV AX,0001
0103 B91000      MOV CX,0010
0106 40          INC AX
0107 49          DEC CX
0108 75FC        JNZ 0106

```

- Расчет времени выполнения команд, которые выполняются в цикле:  
INC+DEC+JNZ равно  $(2+2+16)*15+(2+2+4) = 308$  машинных тактов, т. к. 15 раз в CX не 0, а в шестнадцатый раз содержимое регистра CX=0.
- Расчет времени выполнения всей программы:  $308+4+2 = 314$  тактов.  
При тактовой частоте процессора 1 ГГц на выполнение программы потребуется:  $t=314/1000\ 000\ 000 = 314*10^{-9}$  сек.

## 2. Изучите программу, представленную ниже (программа №2):

```

-A100
0100 MOV AX,01      4 такта
0103 MOV CX, 00010  4 такта
0106 INC AX        2 такта
0107 LOOP 106      17/5 тактов
0109

```

- Команда LOOP (управление циклом) позволяет организовать циклы типа FOR. Команда работает только с регистром CX. Значение в CX служит счетчиком цикла, поэтому его нельзя изменять командами тела цикла.

Работа команды LOOP:

- 1) по команде LOOP содержимое регистра CX уменьшается на 1;
- 2) проверяется CX на равенство нулю;
- 3) если содержимое CX не нуль, то управление передается по указанному адресу, иначе – команде, следующей за LOOP. В программе №2 передается управление команде по адресу 106, команде INC AX.

- После запуска программы (-G109) получен следующий результат:  
**AX=0011 BX=0000 CX=0000 DX=0000**

- Командой U 100,109 выведена на экран программа с адреса 100 до адреса 109:

-U 100,109

0100 B80100           MOV AX,0001

0103 B91000           MOV CX,0010

0106 40                INC AX

0107 E2FD             LOOP 0106

- Расчет времени выполнения команд, которые выполняются в цикле:  
 $INC + LOOP = (2 + 17) * 15 + (2 + 5)$ , где 17 тактов процессор тратит на работу одного прохода цикла и 5 тактов на выход из цикла.
- Расчет времени выполнения всей программы:  
 $4 + 4 + (2 + 17) * 15 + (2 + 5) = 300$ . При тактовой частоте процессора 1 ГГц на выполнение программы потребуется:  $t = 300 / 1000000000 = 300 * 10^{-9} \text{с}$  (1 Гц = 1/сек; 1 ГГц = 1000000000 Гц).
- Сравните время выполнения программ №1 и №2.

**2.1. Разработайте программу вычисления 5!, используя команду Loop.**

2.2. Найдите значение выражения, согласно Вашему варианту, используя команду Loop:

Вариант	Выражение	Вариант	Выражение
1	$Y = \left( \sum_{i=41}^{55} i \right) / 5$	8	$Y = \sum_{i=2}^7 i - 2$
2	$Y = 5300 + \sum_{i=51}^{60} i$	9	$Y = \sum_{i=9}^{76} i + 10$
3	$Y = \sum_{i=3}^9 i + 123$	10	$Y = 8 + \sum_{i=21}^{30} i$
4	$Y = \sum_{i=12}^{21} i - 17$	11	$Y = \left( \sum_{i=3}^{55} i \right) / 10$
5	$Y = 40 - \sum_{i=5}^{11} i$	12	$Y = \sum_{i=4}^{16} i + 3$
6	$Y = \left( \sum_{i=1}^{10} i \right) / 2 + 1$	13	$Y = \sum_{i=2}^{11} i + 4$
7	$Y = 256 - \sum_{i=21}^{30} i$	14	$Y = 1287 + \sum_{i=81}^{93} i$

## СТЕК

Стек представляет собой последовательно расположенные ячейки памяти.

В стек могут быть загружены: содержимое регистра **общего назначения**; содержимое **сегментного регистра**; **данные из памяти**; содержимое **регистра признаков** (флагов, FLAGS).

Глубина стека – количество ячеек памяти, выделенное для стека. «Верхушка» стека – это последняя занятая ячейка в оперативном запоминающем устройстве.

Как правило, стек микропроцессора заполняется в направлении уменьшения адресов ячеек памяти, а регистр-указатель стека **SP** указывает непосредственно «верхнюю» ячейку стека. Машинное слово, являющееся верхним элементом стека, может быть извлечено из него и загружено в регистр, в память или в регистр признаков (флагов, FLAGS).

Программа `debug.exe` создает свой стек, начиная с адреса **FFEE**, который она загружает в регистр **SP** (указатель стека). Этот процесс называется *инициализацией стека*.

В системе команд микропроцессоров Intel XX86 имеются команды для работы со стеком: **PUSH** и **POP**.

**PUSH** - загрузка данных в стек. Формат команды: **PUSH <операнд>**.

**POP** - содержимое ячейки (верхушки стека) помещается по указанному адресу. Формат команды: **POP <адрес>**.

**Пример 1.** Работа со стеком. Базовый способ адресации.

*Дано:*

структуры данных:

0090 dw AAAA

программа:

0100 MOV BX,90

**0103 PUSH [BX]**- Загружается содержимое ячейки 0090 оперативной памяти в стек (базовый способ адресации). Указатель стека, регистр **SP** станет равным FFEC (FFEE - 2 = FFEC)

**Пример 2.** Работа со стеком. Косвенный способ адресации

*Дано:*

структуры данных:

0090 dw AAAA

программа:

**0100 PUSH [90]** - загружается содержимое ячейки 0090 оперативной памяти в стек (косвенный способ адресации).

- Командой `-D FFEC,FFEE` выводится на экран содержимое стека:

`-DFFEC,FFEE`

`XXXX: FFEC`

`AA AA 00`

### Пример 3. Работа со стеком. Косвенный способ адресации

*Дано:*

структуры данных:

0090 dw 16D5

0092 dw 02E3

программа:

**0100 PUSH [90]**

**0104 PUSH [92]**

**0108 POP AX** - извлекается содержимое ячейки стека с адресом FFEA («верхушка стека») в регистр AX.

**0109**

- Результат запуска программы на выполнение с адреса 109; состояние регистров общего назначения:

**AX=02E3 BX=0000 CX=0000 DX=0000 SP=FFEC BP=0000 SI=0000 DI=0000**

**DS=0B72 ES=0B72 SS=0B72 CS=0B72 IP=0109 NV UP EI PL NZ NA PO NC**

**0B72:0109 4E DEC SI**

- Извлечение содержимого из ячейки стека, и запись этого значения в ячейку памяти:

**0109 POP [94]** - извлекается содержимое ячейки стека с адресом FFEC, и записывается в ячейку памяти с адресом 0094;

- обратите внимание на результат работы всей программы:

0090 dw 16D5

0092 dw 02E3

**0100 PUSH [90]**

**0104 PUSH [92]**

**0108 POP AX**

**0109 POP [94]**

**010D**

- содержимое регистров после запуска программы на выполнение с адреса 010D:

```
AX=02E3 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B72 ES=0B72 SS=0B72 CS=0B72 IP=010D NU UP EI PL NZ NA PO NC
0B72:010D C3 RET
```

- регистр SP (указатель стека) принял исходное состояние (SP=FFEE), так как вначале командами по адресам 0100 и 0104, в стек были помещены два значения 16D5 и 02E3 (из ячеек памяти 90 и 92). Значение указателя стека при этом изменялось с FFEE на FFEC и с FFEC на FFEA. Затем, командами по адресам 108 и 109 извлекались из стека значения в регистр AX (значение указателя стека изменилось с FFEA на FFEC) и в ячейку памяти 94 (значение указателя стека изменилось с FFEC на FFEE).

- содержимое памяти:

- ячейки 90 и 92, их значения не изменились;
- ячейка 94, ее значение стало равным 16D5:

```
-D 94
0B72:0090 D5 16 A5 20-A4 AB EF 20 A1 A2 A7 AE
0B72:00A0 A2 AE A3 AE 20 AF AE E0-E2 A0 20 A2 A2 AE A4 A0
0B72:00B0 2D A2 EB A2 AE A4 A0 20-53 42 2C 20 AD A0 AF E0
```

## ПОДПРОГРАММЫ

Работа с подпрограммами осуществляется с помощью команд **CALL** и **RET**.

Команда **CALL** загружает в стек адрес следующей за ней команды. Адрес следующей команды – это содержимое регистра **IP** или регистров **IP** и **CS** в зависимости от формата команды **CALL** (см. систему команд). Адресную часть команда **CALL** использует для изменения содержимого регистра **IP** (или регистров **IP** и **CS**), передавая тем самым управление подпрограмме («вызывает» подпрограмму).

Команда **RET** является последней командой *подпрограммы*. Она извлекает из стека адрес команды и загружает его в регистр **IP** (или регистры **IP** и **CS**), т.е. возвращает управление вызывающей программе.

**Пример 4.** Передача управления в подпрограммах.

Разработать программу записи в регистры **AX**, **BX**, **CX** чисел: 1B5C; 4;5. Запись чисел в регистры **BX** и **CX** выполняют две подпрограммы. Программа располагается по адресу 0100, подпрограммы располагаются, начиная с адреса 0200 в текущем сегменте кода.

### Решение:

```
0100 MOV AX, 1B5C - загружает в регистр AX целое число 1B5C
0103 CALL 200    - обращение к подпрограмме загрузки в регистр CX – числа 4
0106 CALL 204    - обращение к подпрограмме загрузки в регистр BX – числа 5
0109
```

```
0200 MOV CX,4
0203 RET        - возврат управления в основную программу, к адресу 0106
0204 MOV BX,5
0207 RET        - возврат управления в основную программу, к адресу 0109
0208
```

- Содержимое регистров после запуска на выполнение программы с адреса 0109:  
**AX=1B5C BX=0005 CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000**  
**DS=0B72 ES=0B72 SS=0B72 CS=0B72 IP=0106 NV UP EI PL NZ NA PO NC**  
0B72:0106 E8FB00 CALL 0204

- расчет *адреса* перехода к подпрограмме - значения 200 (**CALL 200**):  
Адрес перехода вычисляется как сумма содержимого регистра **IP** и *смещения* в команде **CALL 200**:  
Адрес перехода = **IP** + смещение

**IP**=0106

Расчет *Смещения*:

Вывести на экран программу командой – U100, 109 и определить код команды **CALL 200**:

```
0103 E8FA00 CALL 200
```

Код команды равен: E8FA00 (вызов подпрограммы внутрисегментный с прямой адресацией).

В коде команды E8FA00:

E8 – код операции;

FA – младший байт *смещения*;

00 – старший байт *смещения*.

*Смещение* равно: 00FA.

Адрес перехода = **IP** + *Смещение* = 0106 + 00FA = 0200

## Задания для выполнения в аудитории

### Подпрограммы

**Задание № 1.** Разработать программу для записи в стек содержимого ячеек оперативной памяти (адреса: 0090 ... 0098), используя базовый способ адресации.

Для решения данной задачи выполните следующую последовательность действий:

1. Загрузите последовательность чисел в ячейки оперативной памяти, начиная с адреса 0090:

```
-A90
0090 dw AAAA
0092 dw BBBB
0094 dw 4444
0096 dw 5555
0098 dw 7777
```

2. Введите текст программы, начиная с адреса 100:

- A100

```
0100 MOV BX,90
0103 PUSH [BX]
0105 PUSH [BX+2]
0108 PUSH[BX+4]
010B PUSH[BX+6]
010E PUSH[BX+8]
0111
```

3. Запустите программу на выполнение с адреса 0111 (G 0111). Обратите внимание на состояние регистров:

AX=0000 BX=0090 CX=0000 DX=0000 SP=FFE4 BP=0000 SI=0000 DI=0000

4. Выведите на экран состояние стека после выполнения программы: -d FFE4, FFE4 и обратите внимание на полученный результат:

FFE0        77 77 55 55-44 44 BB BB AA AA 00

5. Сохраните текст разработанной Вами программы( для выполнения СРС).

### Задание № 2.

- Разработайте программу, которая загружает данные, размещенные в ячейках памяти (адреса с 0090 по 0098) в стек, после чего извлекает содержимое ячеек стека и размещает извлеченные данные в регистрах общего назначения. Для обращения к ячейкам памяти используйте *косвенный* способ адресации. Адреса ячеек памяти для размещения исходных данных и регистры используйте согласно Вашему варианту. Например, содержимое ячеек 90, 92, 94 должно быть записано в стек, и из стека извлечено для записи в регистры AX, DX, CX соответственно.



Вариант	Адреса ячеек памяти	Регистры общего назначения	Вариант	Адреса ячеек памяти	Регистры общего назначения
1	92, 94, 96	AX, BX, CX	8	92, 94, 96	AX, DX, CX
2	90, 92, 94	BX, CX, DX	9	94, 96, 98	DX, BX, AX
3	94, 96, 98	CX, BX, AX	10	92, 94, 96	BX, CX, DX
4	92, 94, 96	DX, AX, CX	11	90, 92, 94	CX, AX, BX
5	90, 92, 94	AX, DX, BX	12	94, 96, 98	AX, BX, DX
6	94, 96, 98	BX, CX, AX	13	92, 94, 96	BX, DX, AX
7	90, 92, 94	DX, BX, CX	14	90, 92, 94	DX, BX, CX

- Выведите на экран состояние стека после выполнения программы.
- Сохраните текст разработанной Вами программы (для выполнения СРС).

### Задание № 3.

- Разработайте программу, с помощью которой выполняются следующие действия:

1. запись чисел **9887**, **B** и **C** в ячейки памяти **0092**, **0094** и **0096** соответственно.
2. вызов подпрограмм №1, 2, 3.

Подпрограммы расположены, начиная с адреса **200**, и выполняют следующие действия:

- подпрограмма №1:
    - пересылает в регистр **AX** значение **9887**;
    - пересылает в регистр **DX** значение **0B**;
  - подпрограмма №2:
    - увеличивает значение регистра **DX** на 3;
  - подпрограмма №3:
    - суммирует содержимое регистра **AX** с содержимым ячейки **0092** оперативной памяти;
    - вызывает подпрограмму №4;
    - пересылает в регистр **BX** значение, хранящееся в ячейке **0094**;
    - вычитает из содержимого регистра **CX** содержимое регистра **DX**;
  - подпрограмма №4:
    - пересылает в регистр **CX** значение, хранящееся в ячейке **0096**;
- Сохраните текст разработанной Вами программы ( для выполнения СРС).

## Задания для самостоятельной работы студента (СРС)

---

**Задание № 1.** Используя текст программы из задания №1 аудиторной работы, укажите значение указателя стека (регистр SP) и состояние стека после выполнения каждой команды программы, заполнив следующую таблицу:

Команды программы	Пояснения	Значение указателя стека (регистр SP)	Состояние стека
0100 MOV BX,90	в регистр BX загружается значение базы	FFEE	0000
...	...	...	...

**Задание № 2.** Выполните задание №2 из аудиторной работы, *согласно Вашему варианту*, используя цикл LOOP и самостоятельную инициализацию стека (инициализация стека пользователем). Запишите команду вывода на экран содержимого стека, при указанной инициализации.

**Задание № 3.** Для программы из задания №3 аудиторной работы:

**3.1.** Рассчитайте *адрес* перехода к подпрограмме №3;

**3.2.** Заполните таблицу, указав следующую информацию:

Адрес	Коды команд	Мнемоника команд	Адрес указателя стека (после выполнения команды)	Содержимое стека (после выполнения команды)

1. Гибсон Г., Лю Ю-Чжен. Микропроцессоры семейства 8086/8088. Архитектура, программирование и проектирование микрокомпьютерных систем/ пер. с англ. М.: Радио и связь, 1987. 512 с.
2. Гук М. Процессоры Intel: от 8086 до Pentium II. СПб.: Питер, 1998. 224 с.
3. FE23 LOOCH DISASM СПРАВОЧНИК ПО КОМАНДАМ ПРОЦЕССОРОВ x86. URL: <http://looch-disasm.narod.ru/refe20.htm>.
4. Режимы адресации 16-разрядного микропроцессора Intel-8086. URL: <http://studopedya.ru/1-40528.html>.
5. Студопедия. Первые процессоры. URL: [https://studopedia.ru/10\\_160778\\_pervie-protssessori.html](https://studopedia.ru/10_160778_pervie-protssessori.html).
6. О разрядности процессоров. URL: <https://www.ixbt.com>.

Приложение 1

Система команд микропроцессора Intel 8086

Таблица 1

Мнемоника	Описание	Число тактов	Число байт	Число обращений
AAA	ASCII коррекция для сложения	4	1	–
AAD	ASCII коррекция для деления	60	2	–
AAM	ASCII коррекция для умножения	83	2	–
AAS	ASCII коррекция для вычитания	4	1	–
ADC	Сложить с переносом			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2 – 4	1
M,REG	Память – регистр	16+EA	2 – 4	2
ACC,DATA	Аккумулятор – непосредственный операнд	4	2 – 3	–
R,DATA	Регистр – непосредственный операнд	4	3 – 4	–
M,DATA	Память – непосредственный операнд	17+EA	3 – 6	2
R16,DATA_S	Регистр – регистр	4	3	–
M16,DATA_S	Регистр – память	17+EA	3 – 5	2
<b>ADD</b>	<b>Сложить</b>			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2 – 4	1
M,REG	Память – регистр	16+EA	2 – 4	2
ACC,DATA	Аккумулятор – непосредственный операнд	4	2 – 3	–
R,DATA	Регистр – непосредственный операнд	4	3 – 4	–
M,DATA	Память – непосредственный операнд	17+EA	3 – 6	2
R16,DATA_S		4	3	–
M16,DATA_S		17+EA	3 – 5	2
AND	Объединить по «и» (конъюнкция)			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2 – 4	1
M,REG	Память – регистр	16+EA	2 – 4	2
ACC,DATA	Аккумулятор – непосредственный операнд	4	2 – 3	–
R,DATA	Регистр – непосредственный операнд	4	3 – 4	–
M,DATA	Память – непосредственный операнд	17+EA	3 – 6	2

<b>Мнемоника</b>	<b>Описание</b>	<b>Число тактов</b>	<b>Число байт</b>	<b>Число обращений</b>
CALL	Вызов процедуры			
NEAR_PROC	Внутри сегментный прямой	19	3	1
R16(NEAR)	Внутри сегментный косвенный через регистр	16	2	1
M16(NEAR)	Внутри сегментный косвенный через память	21+EA	2–4	2
FAR_PROC	Межсегментный прямой	28	5	2
M32(FAR)	Межсегментный косвенный	37+EA	2–4	4
CBW	Преобразовать байт в слово	2	1	–
CLC	Сбросить флажок переноса	2	1	–
CLD	Сбросить флажок направления	2	1	–
CLI	Сбросить флажок прерывания	2	1	–
CMC	Инвертировать флажок переноса	2	1	–
CMP	Сравнить			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2–4	1
M,REG	Память – регистр	9+EA	2–4	1
ACC,DATA	Аккумулятор – непосредственный операнд	4	2–3	–
R,DATA	Регистр – непосредственный операнд	4	3–4	–
M,DATA	Память – непосредственный операнд	10+EA	3–6	1
R16,DATA_		4	3	–
M16,DATA_S		10+EA	3–5	1
CMPS /	Сравнить цепочки /	1		
CMPSB /	Сравнить цепочки байт /			
CMPS	Сравнить цепочки слов			
D_STR.S_	без повторения	22	1	2
STR	с повторением	9+22*N	1	2*N
CWD	Преобразовать слово в двойное слово	5	1	–
DAA	Десятичная коррекция для сложения	4	1	–
DAS	Десятичная коррекция для вычитания	4	1	–
DEC	Декремент на 1			
R16	16–битный регистр	2	1	–
R8	8–битный регистр	3	2	–
M	Память	15+EA	2–4	2
DIV	Разделить без знака			
R8	8–битный регистр	80–90	2	–
R16	16–битный регистр	144–162	2	–
M8	8–битная память	(86–96)+EA	2–4	1

Мнемоника	Описание	Число тактов	Число байт	Число обращений
M16	16-битная память	(150 – 168)+EA	2 – 4	1
ESC	Переключиться на сопроцессор			
OPCODE,R	Регистр	2	2	–
OPCODE,ME M	Память	8+EA	2 – 4	1
HLT	Остановить	2	1	–
IDIV	Разделить со знаком			
R8	8-битный регистр	101 – 112	2	–
R16	16-битный регистр	165 – 184	2	–
M8	8-битная память	(107 – 118)+EA	2 – 4	1
M16	16-битная память	(171 – 190)+EA	2 – 4	1
IMUL	Умножить со знаком			
R8	8-битный регистр	80 – 98	2	–
R16	16-битный регистр	128 – 154	2	–
M8	8-битная память	(86 – 104)+EA	2 – 4	1
M16	16-битная память	(134 – 160)+EA	2 – 4	1
IN	Ввести из порта ввода-вывода			
ACC,PORT	Фиксированный порт	10	2	1
ACC,DX	Переменный порт	8	1	1
INC	Инкремент на 1			
R16	16-битный регистр	2	1	–
R8	8-битный регистр	3	2	–
INT	Прерывание			
N_PROC3	Тип = 3	52	1	5
N_PROC	Тип <math>\leq 3</math>	53	2	5
INTO	Прерывание при переполнении			
	Прерывание есть	53	1	5
	Прерывания нет	4	1	5
IRET	Возвратиться из прерывания	24	1	3
JA/	Перейти, если выше/	16/4	2	–
JNBE	Перейти, если не ниже или равно	16/4	2	–
JAE/	Перейти, если выше или равно/	16/4	2	–
JNB/	Перейти, если не ниже/	16/4	2	–
JNC	Перейти, если нет переноса	16/4	2	–
JB/	Перейти, если ниже/	16/4	2	–
JNAE/	Перейти, если не выше или равно	16/4	2	–

<b>Мнемоника</b>	<b>Описание</b>	<b>Число тактов</b>	<b>Число байт</b>	<b>Число обращений</b>
JC	Перейти, если есть перенос	16/4	2	–
JBE/	Перейти, если ниже или равно/	16/4	2	–
JNA	Перейти, если не выше	16/4	2	–
JCXZ	Перейти, если CX равен нулю	18/6	2	–
JE/	Перейти, если равно/	16/4	2	–
JZ	Перейти, если нуль	16/4	2	–
JQ/	Перейти, если больше/	16/4	2	–
JNLE	Перейти, если не меньше или равно	16/4	2	–
JQE/	Перейти, если больше или равно/	16/4	2	–
JNL	Перейти, если не меньше	16/4	2	–
JL/	Перейти, если меньше/	16/4	2	–
JNQE	Перейти, если не больше или равно	16/4	2	–
JLE/	Перейти, если меньше или равно/	16/4	2	–
JNQ	Перейти, если не больше	16/4	2	–
JMP	Переход			
SHORT_TAG	Внутрисегментный прямой короткий	15	2	–
NEAR_TAG	Внутрисегментный прямой	15	3	–
R16(NEAR)	Внутрисегментный косвенный через регистр	11	2	–
M16(NEAR)	Внутрисегментный косвенный через память	18+EA	2 – 4	1
FAR_TAG	Межсегментный прямой	15	5	–
M32(FAR)	Межсегментный косвенный	24+EA	2 – 4	2
JNE/	Перейти, если не равно/	16/4	2	–
JNZ	Перейти, если не нуль	16/4	2	–
JNO	Перейти, если нет переполнения	16/4	2	–
JNP/	Перейти, если нет паритета/	16/4	2	–
JPO	Перейти, если паритет нечетный	16/4	2	–
JNS	Перейти, если нет знака	16/4	2	–
JO	Перейти, если есть переполнение	16/4	2	–
JP/	Перейти, если есть паритет/	16/4	2	–
JPE	Перейти, если паритет четный	16/4	2	–
JS	Перейти, если есть знак	16/4	2	–
LAHF	Загрузить AH из флажков	4	1	–
LDS	Загрузить указатель, используя DS	16+EA	2 – 4	2
LEA	Загрузить эффективный адрес	2+EA	2 – 4	–
LES	Загрузить указатель, используя ES	16+EA	2 – 4	2
LOCK	Заблокировать шину	2	1	–

<b>Мнемоника</b>	<b>Описание</b>	<b>Число тактов</b>	<b>Число байт</b>	<b>Число обращений</b>
LODS/	Загрузить цепочку/	12	1	1
LODSB/	Загрузить цепочку байт/	12	1	1
LODSW	Загрузить цепочку слов			
D_STR.S_	без повторения	12	1	1
STR	с повторением	9+13*N	1*N	1
LOOP	Зациклить	17/5	2	–
LOOPE/	Зациклить, если равно/	18/6	2	–
LOOPNE/	Зациклить если не равно/	19/5	2	–
LOOPNZ	Зациклить, если не ноль	19/5	2	–
LOOPZ	Зациклит, если ноль	18/6	2	–
<b>MOV</b>	<b>Переслать</b>			
SIMP_M, ACC	Память – аккумулятор	10	3	1
ACC,SIMP_M	Аккумулятор – память	10	3	1
SR,R16	Регистр – регистр	2	2	–
SR,M16	Память – регистр	8+EA	2 – 4	1
R16,SR	Сегментный регистр – регистр	2	2	–
M16,SR	Сегментный регистр – память	9+EA	2 – 4	1
REG,R	Регистр – регистры SS,DS или ES	2	2	–
REG,M	Память – регистры SS,DS или ES	8+EA	2 – 4	1
M,REG	Регистр – память	9+EA	2 – 4	1
REG,DATA	Непосредственный операнд – регистр	4	2 – 3	–
M,DATA	Непосредственный операнд – память	10+EA	3 – 6	1
MOVS	Переслать цепочку			
D_STR.S_	без повторения	18	1	2
S_STR	с повторением	9+17/повтор		
MOVSB/	Загрузить цепочку байт	9+17*N	1	2*N
MOVSW	Загрузить цепочку слов	9+17*N	1	2*N
<b>MUL</b>	<b>Умножить без знака</b>			
R8	8–битный регистр	70 – 77	2	–
R16	16–битный регистр	118 – 133	2	–
M8	8–битная память	(76 – 83)+EA	2 – 4	1
M16	16–битная память	(124–139)+EA	2 – 4	1
NEG	Изменить знак			
R	Регистр	3	2	–
M	Память	16+EA	2 – 4	2



<b>Мнемоника</b>	<b>Описание</b>	<b>Число тактов</b>	<b>Число байт</b>	<b>Число обращений</b>
NOP (XCHG AX,AX)	Холостая команда	3	1	–
NOT	Инвертировать (отрицание)			
R	Регистр	3	2	–
M	Память	16+EA	2–4	2
OR	Объединить по «или» (дизъюнкция)			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2–4	1
M,REG	Память – регистр	16+EA	2–4	2
ACC,DATA	Непосредственный операнд – аккумулятор	4	2–3	–
R,DATA	Непосредственный операнд – регистр	4	3–4	–
M,DATA	Непосредственный операнд – память	17+EA	3–6	2
OUT	Вывести в порт ввода – вывода			
PORT,ACC	Фиксированный порт	10	2	1
DX,ACC	Переменный порт	8	1	1
POP	Извлечь слово из текста			
REG16	В регистр	8	1	1
M16	Память	17+EA	2–4	2
SR	Сегментный регистр	8	1	1
POPF	Извлечь флажки из стека	8	1	1
PUSH	Включить слово в стек			
REG16	Регистр	11	1	1
M16	Память	16+EA	2–4	2
SR	Сегментный регистр	10	1	1
PUSHF	Включить флажки в стек	10	1	1
RCL	Циклически сдвинуть влево через перенос			
R,VALI	Регистр (сдвиг на один бит)	2	2	–
M,VALI	Память (сдвиг на один бит)	15+EA	2–4	2
R,CL	Регистр (переменный сдвиг)	8+4*N	2	–
M,CL	Память (переменный сдвиг)	20+EA+4*N	2–4	2
RCR	Циклически сдвинуть вправо через перенос			
R,VALI	Регистр (сдвиг на один бит)	2	2	–
M,VALI	Память (сдвиг на один бит)	15+EA	2–4	2
R,CL	Регистр (переменный сдвиг)	8+4*N	2	–
M,CL	Память (переменный сдвиг)	20+EA+4*N	2–4	2
RET	Возврат из процедуры			

<b>Мнемоника</b>	<b>Описание</b>	<b>Число тактов</b>	<b>Число байт</b>	<b>Число обращений</b>
(NEAR)	Внутрисегментный	8	1	1
POP_VAL (NEAR)	Внутрисегментный с константой	12	3	1
(FAR)	Межсегментный	18	1	2
POP_VAL (FAR)	Межсегментный мс константой	17	3	2
ROL	Циклически сдвинуть влево			
R,VALI	Регистр (сдвиг на один бит)	2	2	–
M,VALI	Память (сдвиг на один бит)	15+EA	2 – 4	2
R,CL	Регистр (переменный сдвиг)	8+4*N	2	–
M,CL	Память (переменный сдвиг)	20+EA+4*N	2 – 4	2
ROR	Циклически сдвинуть вправо			
R,VALI	Регистр (сдвиг на один бит)	2	2	–
M,VALI	Память (сдвиг на один бит)	15+EA	2 – 4	2
R,CL	Регистр (переменный сдвиг)	8+4*N	2	–
M,CL	Память (переменный сдвиг)	20+EA+4*N	2 – 4	2
SAHF	Заполнить AH во флажках	4	1	–
SAL	Сдвинуть арифметически влево			
R,VALI	Регистр (сдвиг на один бит)	2	2	–
M,VALI	Память (сдвиг на один бит)	15+EA	2 – 4	2
R,CL	Регистр (переменный сдвиг)	8+4*N	2	–
M,CL	Память (переменный сдвиг)	20+EA+4*N	2 – 4	2
SHL	Сдвинуть логически влево			
R,VALI	Регистр (сдвиг на один бит)	2	2	–
M,VALI	Память (сдвиг на один бит)	15+EA	2 – 4	2
R,CL	Регистр (переменный сдвиг)	8+4*N	2	–
M,CL	Память (переменный сдвиг)	20+EA+4*N	2 – 4	2
SAR	Сдвинуть арифметически вправо			
R,VALI	Регистр (сдвиг на один бит)	2	2	–
M,VALI	Память (сдвиг на один бит)	15+EA	2 – 4	2
R,CL	Регистр (переменный сдвиг)	8+4*N	2	–
M,CL	Память (переменный сдвиг)	20+EA+4*N	2 – 4	2
SHR	Сдвинуть логически вправо			
R,VALI	Регистр (сдвиг на один бит)	2	2	–
M,VALI	Память (сдвиг на один бит)	15+EA	2 – 4	2

<b>Мнемоника</b>	<b>Описание</b>	<b>Число тактов</b>	<b>Число байт</b>	<b>Число обращений</b>
R,CL	Регистр (переменный сдвиг)	8+4*N	2	–
M,CL	Память (переменный сдвиг)	20+EA+4*N	2 – 4	2
SBB	Вычитать с заемом			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2 – 4	1
M,REG	Память – регистр	16+EA	2 – 4	2
ACC,DATA	Непосредственный операнд – аккумулятор	4	2 – 3	–
R,DATA	Непосредственный операнд – регистр	4	3 – 4	–
M,DATA	Непосредственный операнд – память	17+EA	3 – 6	2
R16,DATA_S		4	3	–
M16,DATA_S		17+EA	3 – 5	2
SCAS	Сканировать цепочку	15	1	1
SCASB	Сканировать цепочку байт	9+15*N	1	1*N
SCASW	Сканировать цепочку слов	9+15*N	1	1*N
STC	Установить флажок переноса	2	1	–
STD	Установить флажок направления	2	1	–
STI	Установить флажок прерывания	2	1	–
STOS	Запомнить цепочку	11	1	1
STOSB	Запомнить цепочку байт	9+10*N	1	1*N
STOSW	Запомнить цепочку слов	9+10*N	1	1*N
<b>SUB</b>	<b>Вычитать</b>			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2 – 4	1
M,REG	Память – регистр	16+EA	2 – 4	2
ACC,DATA	Непосредственный операнд – аккумулятор	4	2 – 3	–
R,DATA	Непосредственный операнд – регистр	4	3 – 4	–
M,DATA	Непосредственный операнд – память	17+EA	3 – 6	2
R16,DATA_S		4	3	–
M16,DATA_S		17+EA	3 – 5	2
TEST	Проверить			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2 – 4	1
M,REG	Память – регистр	9+EA	2 – 4	1
ACC,DATA	Непосредственный операнд – аккумулятор	4	2 – 3	–
R,DATA	Непосредственный операнд – регистр	5	3 – 4	–

Мнемоника	Описание	Число тактов	Число байт	Число обращений
M,DATA	Непосредственный операнд – память	11+EA	3 – 6	1
WAIT	Ожидать активного сигнала TEST	3+5*N	1	–
XCHG	Обменять			
AX,REG16	Аккумулятор – регистр	3	1	–
REG16,AX	Регистр – аккумулятор	3	1	–
REG,R	Регистр – регистр	4	2	–
REG,M	Регистр – память	17+EA	2 – 4	2
M,REG	Память – регистр	17+EA	2 – 4	2
XLAT/XLATB	Преобразовать	11	1	1
XOR	Сложить по модулю 2 (исключающее – или)			
REG,R	Регистр – регистр	3	2	–
REG,M	Регистр – память	9+EA	2 – 4	1
M,REG	Память – регистр	16+EA	2 – 4	2
ACC,DATA	Непосредственный операнд – аккумулятор	4	2 – 3	–
R,DATA	Непосредственный операнд – регистр	4	3 – 4	–
M,DATA	Непосредственный операнд – память	17+EA	3 – 6	2

Таблица 2

Значение поля R/M	MOD = 00		MOD = 01		MOD = 10	
		Время вычисления EA (в тактах)		Время вычисления EA (в тактах)		Время вычисления EA (в тактах)
000	$(BX)+(SI)$	7	$(BX)+(SI)+DISP8$	11	$(BX)+(SI)+DISP16$	11
001	$(BX)+(DI)$	8	$(BX)+(DI)+DISP8$	12	$(BX)+(DI)+DISP16$	12
010	$(BP)+(SI)$	8	$(BP)+(SI)+DISP8$	12	$(BP)+(SI)+DISP16$	12
011	$(BP)+(DI)$	7	$(BP)+(DI)+DISP8$	11	$(BP)+(DI)+DISP16$	11
100	$(SI)$	5	$(SI)+DISP8$	9	$(SI)+DISP16$	9
101	$(DI)$	5	$(DI)+DISP8$	9	$(DI)+DISP16$	9
110	$DISP16$	6	$(BP)+DISP8$	9	$(BP)+DISP16$	9
111	$(BX)$	5	$(BX)+DISP8$	9	$(BX)+DISP16$	9

Команды условных переходов

Команда	Условие	Условие
JA	$X > Y$	$CF = 0 \ \& \ ZF = 0$
JAЕ	$X \geq Y$	$CF = 0$
JB	$X < Y$	$CF = 1$
JBE	$X < Y$	$CF = 1 \ \text{or} \ ZF = 1$
JC		$CF = 1$
JCXZ		$CX = 0$
JE (тоже, что и JZ)	$X = Y$	$ZF = 1$
JG	$X > Y$	$CF = 0 \ \& \ SF = 0$
JGE	$X \geq Y$	$SF = OF$
JL	$X < Y$	$SF \neq OF$
JLE	$X \leq Y$	$ZF = 1 \ \text{or} \ SF \neq OF$
JNA	$X \leq Y$	$CF = 1 \ \text{or} \ ZF = 1$
JNAE	$X < Y$	$CF = 1$
JNB	$X \geq Y$	$CF = 0$
JNBE	$X > Y$	$CF = 1 \ \& \ ZF = 0$
JNC		$CF = 0$
JNE	$X \neq Y$	$ZF = 0$
JNG	$X \leq Y$	$ZF = 1 \ \text{or} \ SF \neq OF$
JNGE	$X < Y$	$SF \neq OF$
JNL	$X \geq Y$	$SF = OF$
JNLE	$X > Y$	$ZF = 0 \ \& \ SF = OF$
JNO		$OF = 0$
JNP		$PF = 0$
JNS		$SF = 0$
JNZ	$X \neq Y$	$ZF = 0$
JO		$OF = 1$
JP		$PF = 1$
JPE		$PF = 1$
JPO		$PF = 1$
JS		$SF = 1$
JZ		$Z = 1$

Приложение 3

**Вспомогательные регистры: (флаговый регистр) – набор отдельных регистров-битов (флагов)**

	Наименование флага	Назначение	Обозначение	Значение, когда установлен	Значение, когда сброшен
<b>1. СТАТУСНЫЕ ФЛАГИ</b>					
1.1	Флаг переноса (да / нет)	Указывает на наличие переноса при выполнении арифметических операций	<b>CF</b>	СУ	NC
1.2	Флаг переполнения (да / нет)	Указывает на переполнение при выполнении арифметических операций	<b>OF</b>	OV	NV
1.3	Флаг нуля (да / нет)	Указывает на нулевой результат при выполнении арифметических операций	<b>ZF</b>	ZR	NZ
1.4	Флаг знака (отриц. / положит.)	Указывает на отрицательный результат при выполнении арифметических операций или устанавливает признак неравенства при операциях сравнения	<b>SF</b>	NG	PL
1.5	Флаг четности (четное / нечетное)	Указывает на наличие в операнде четного <b>числа битов</b> , равных 1	<b>PF</b>	PE	PO
1.6	Флаг арифметического переноса (да / нет)	Указывает на необходимость корректировки после выполнения арифметических операций с числами, представленных в виде двоично-десятичных кодов	<b>AF</b>	AC	NA
<b>2. УПРАВЛЯЮЩИЕ ФЛАГИ</b>					
2.1	Флаг направления (уменьшение / увеличение)	Управляет способом изменения адресов при операциях побайтовой обработке данных, например при работе со строками	<b>DF</b>	DN	UP
2.2	Флаг прерывания (возможно / невозможно)	Управляет процессом блокировки и разблокировки прерываний	<b>IF</b>	EI	DI
2.3	Флаг трассировки	Управляет пошаговым выполнением операций	<b>TF</b>		

*Учебное издание*

**Ромашкина Татьяна Витальевна**

**Миндоров Николай Иванович**

**ИНФОРМАТИКА  
И ОСНОВЫ ПРОГРАММИРОВАНИЯ**

Учебное пособие

Редактор *М. А. Шемякина*

Техническая подготовка материалов: *Т. В. Ромашкина*

---

Объем данных 3 Мб

Подписано к использованию 09.11.2018

---

Размещено в открытом доступе  
в электронной мультимедийной библиотеке  
ELiS: <https://elis.psu.ru>

Издательский центр  
Пермского государственного  
национального исследовательского университета  
614990, г. Пермь, ул. Букирева, 15