



ЭКЗАМЕНАЦИОННЫЕ ВОПРОСЫ

ПОДГОТОВИЛ БАШМАШНИКОВ ИЛЬЯ

ВОПРОС 1: ПРЕДСТАВЛЕНИЯ ЗНАКОВЫХ ЦЕЛЫХ ЧИСЕЛ И ОСНОВНЫЕ ОПЕРАЦИИ НАД НИМИ. ПРИМЕРЫ

Целые числа - это простейшие числовые данные, с которыми оперирует ЭВМ. Для целых чисел существуют два представления: беззнаковое (только для неотрицательных целых чисел) и со знаком. Отрицательные числа можно представлять только в знаковом виде. Целые числа в компьютере хранятся в формате с фиксированной запятой.

Представление знаковых целых чисел

Знаковые целые числа в памяти компьютера представляются с помощью *дополнительного кода*.

Вычисление дополнительного кода числа x :

1. Если $x \geq 0$, то дополнительный код x совпадает с его двоичным представлением

2. Если $x < 0$, то для получения дополнительного кода необходимо:

A. найти двоичное представление модуля x

B. Изменить все биты в двоичном представлении x на противоположные

C. Прибавить единицу

Например, представим число -84 в знаковом однобайтовом типе (тип char в C):

A. $|-84_{10}| = 84_{10} = 2^6 + 2^4 + 2^2 = 01010100_2$

B. Инvertируем все биты: $01010100_2 \rightarrow 10101011_2$

C. Прибавляем единицу: $10101011_2 + 1_2 = 10101100_2$

7 0

1	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---

Целые числа со знаком (прямой код)

Для кодирования **целых чисел со знаком** отводят 8, 16, 32 или 64 бита.

Старший разряд числа определяет его знак.

Если он равен 0, число положительное, если 1, то отрицательное.

$$53_{10} = 110101_2$$

0	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

$$-53_{10} = -110101_2$$

1	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

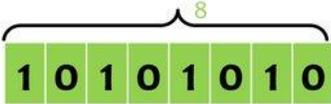
Такое представление чисел в компьютере называется **прямым** кодом.

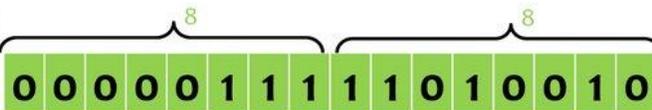
ВОПРОС 2: ПРЕДСТАВЛЕНИЕ ДВОИЧНЫХ ЧИСЕЛ В ФОРМЕ С ФИКСИРОВАННОЙ ЗАПЯТОЙ

В вычислительных машинах применяется форма представления с **фиксированной запятой** все числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной

32

Числа с фиксированной точкой («,»)

$A_2 = 10101010_2$ 

$2002_{10} = 11111010010_2$ 

$\pi = 3,14159265359$

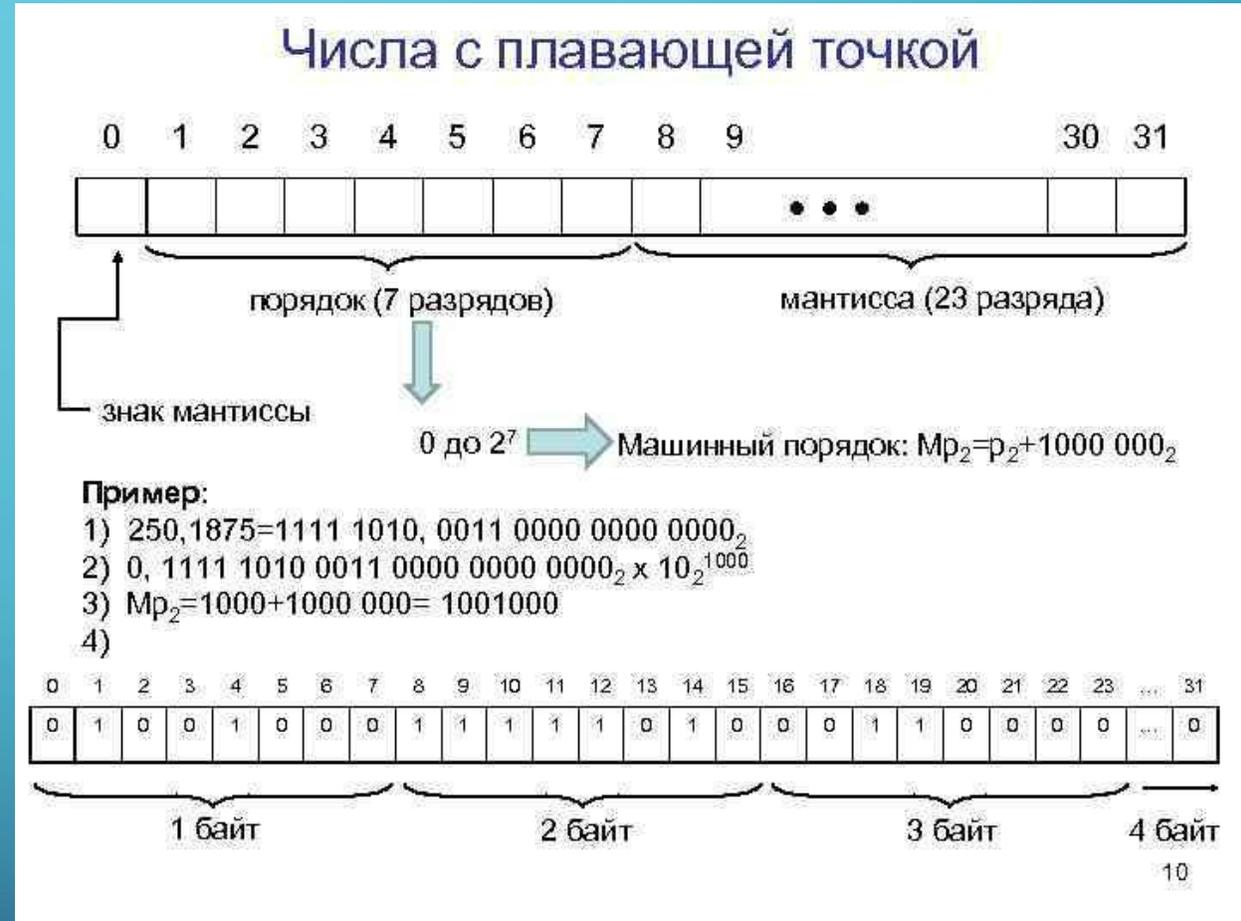
$3 = 11_2$

$0,14 = 0,00100011_2$

Знак	Целая часть	Дробная часть
0	0000011	00100011

ВОПРОС 3: ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ФОРМАТЕ С ПЛАВАЮЩЕЙ ТОЧКОЙ И ОСНОВНЫЕ ОПЕРАЦИИ НАД НИМИ

В форме представления с чисел плавающей запятой каждое число изображается в виде двух групп цифр. Первая группа цифр называется мантиссой, вторая – порядком, причем абсолютная величина мантиссы должна быть меньше 1, а порядок – целым числом



ВОПРОС 4: ОРГАНИЗАЦИЯ ХРАНЕНИЯ И РАБОТЫ С ДВУМЕРНЫМИ МАССИВАМИ. ДВУМЕРНЫЕ МАССИВЫ

Многомерный массив это массив, элементами которого служат массивы. Многомерные массивы задаются указанием каждого измерения в квадратных скобках

Массивы Двумерные массивы



- Двумерный массив (матрица) – это прямоугольная таблица, состоящая из нескольких строк и столбцов.

```
-1  3  -2  6
-4  0   5 -2
 9 -8   3  2
```

Обозначение: $a[i,j]$,
где i – номер строки,
 j – номер столбца

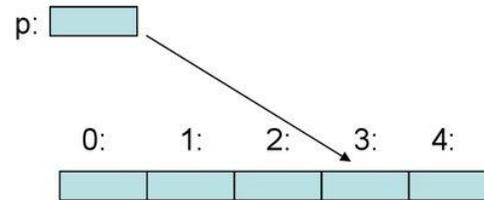
```
var  
    a:array[1..n, 1..m] of integer;
```

ВОПРОС 5:
ПОНЯТИЕ УКАЗАТЕЛЯ.
СТАТИЧЕСКИЕ И
ДИНАМИЧЕСКИЕ ПЕРЕМЕННЫЕ.
ФУНКЦИИ ДЛЯ РАБОТЫ С
ДИНАМИЧЕСКОЙ ПАМЯТЬЮ

Указатель – переменная, значением которой является адрес ячейки памяти. То есть указатель ссылается на блок данных из области памяти, причём на самое его начало. Указатель может ссылаться на переменную или функцию.

Указатели

`char* p;`



Пример:

```
char v[5];
```

```
char* p=&v[3]
```

* - операция «взять содержимое»

& - операция «взять адрес»

Указатель это специальная переменная для хранения адреса другой переменной

СТАТИЧЕСКИЕ И ДИНАМИЧЕСКИЕ ПЕРЕМЕННЫЕ

Сравнение статической и динамической памяти

Параметры сравнения	Статические переменные	Динамические переменные
Способ распределения памяти	Автоматическое (во время компиляции)	Управляется программой
Место расположения	Глобальные переменные – в сегменте кода, локальные – в сегменте стека	В динамической памяти (куче)
Способ доступа	По имени (идентификатор)	По адресу (указатель на место расположения в памяти)

Функции работы с динамической памятью

Функции	Прототипы и краткое описание
malloc	void * malloc (unsigned s); Возвращает указатель на начало области динамической памяти длиной в s байт. При неудачном завершении возвращает значение NULL.
calloc	void * calloc (unsigned n, unsigned m); Возвращает указатель на начало области обнуленной динамической памяти, выделенной для размещения n элементов по m байт каждый. При неудачном завершении возвращает значение NULL.
realloc	void * realloc (void * p, unsigned ns); Изменяет размер блока ранее выделенной памяти до размера ns байт. p - адрес начала изменяемого блока. Если p = NULL (память раньше не выделялась), то функция выполняется как malloc.
free	void * free (void *p); Освобождает ранее выделенный участок динамической памяти, адрес первого байта которого равен значению p.