Массив

Задача: даны 100 целых чисел, нужно найти их среднее значение.

Массив это именованный набор однотипных элементов, расположенных в памяти непосредственно друг за другом, обращаться к которым можно по их номеру.
Индекс это номер элемента в массиве.

Объявление массива:

```
int a[100];
```

Обращение к элементу массива:

```
a[1]=10;
b=a[2];
```

Пример использования массива

Задача: даны 100 целых чисел, нужно найти их среднее значение.

```
1  #include <time.h>
2  main()
3  {
4      srand(time(0));
5      int i,s=0;
6      int a[100];
7      for(i=0;i<100;i++)
8          a[i]=rand();
9      for(i=0;i<100;i++)
10         s+=a[i];
11     printf("%i",s/100);
12 }
```

Свойства массивов

Индексы элементов начинаются с нуля:

```
int a[10];
a[0]=0;      // Правильно
a[9]=9;      // Правильно
a[10]=10;    // ОШИБКА!
```

Массивы могут содержать элементы любых типов:

```
int a[10];
float b[5];
char c[10];
```

Свойства массивов

Массив можно инициализировать при объявлении:

```
float a[]  = {1, 1.5, 2, 2.5, 3};
```

В С не отслеживаются границы массива!

```
1  main()
2  {
3      int a[10];
4      a[10000]=1;
5  }
```

[] - операция индексирования

Свойства массивов

Массив можно инициализировать при объявлении:

```
float a[]  = {1, 1.5, 2, 2.5, 3};
```

В С не отслеживаются границы массива!

```
1  main()
2  {
3      int a[10];
4      a[10000]=1;
5  }
```

[] - операция индексирования

Массивы в других языках программирования

В языке С не реализовано, но в других языках программирования, или при использовании библиотек, массивы могут:

- контролировать выход за границу массива
- динамически изменять размер массива
- индексировать элементы по имени (ассоциативные массивы)
- выполнять операции над массивами целиком (например, складывать поэлементно)
-и другие действия

Массивы строк

Строкова константа

```
char a[] = “Привет”;
char a[] = {'П', 'р', 'и', 'в', 'е', 'т', '\0'};
```

Массив строк заканчивающихся нулем можно распечатать:

```
printf(“%s”, a);
```

Библиотека string.h содержит функции для работы с массивами строк. Например:

```
strcpy(dest, scr);
```

Структуры

Структура это именованный набор данных *различных* типов.

Каждый элемент структуры имеет имя, по которому к нему можно обратиться.

Объявление структуры:

```
struct tovar
{
    char name[20];
    int price;
};
```

Объявление переменных:

```
tovar a,b;
```

Операции со структурами

Инициализация структуры:

```
tovar a={"мыло", 100};
```

Обращение к полю структуры:

```
a.price = 10;
p = a.price;
```

Над структурами и массивами нельзя выполнять операции присваивания, сложения, вычитания и другие. Эти операции можно выполнять только над элементами массивов и структур.

Задание к лабораторной работе

1 Создать массив из 50 чисел, значения которых равны значению функции от индекса элемента. Функции взять из второго задания предыдущей лабораторной работы. Распечатать значения этого массива.

Задание к лабораторной работе

2 Пользователь вводит с клавиатуры слово (не больше 19 букв). Его нужно распечатать вертикально, выводя по одной букве в строке. Ввод пользователем слова в массив символов a можно реализовать следующей командой: scanf("%s", a). При выводе строки на экран нужно иметь в виду, что количество символов в строке неизвестно, но известно, что последний символ – '\0'. При написании программы подумайте, как можно сократить количество операций в ней.

Задание к лабораторной работе

3 Объявить структуру, содержащую строки, соответствующие имени и фамилии человека. Создать массив из 10 таких структур, инициализировать их именами и фамилиями любых известных людей. Напечатать список людей, чье имя совпадает с именем, которое ввел пользователь.