

Лабораторная работа №11. Составление программ для одномерных массивов с помощью операторов массива.

Лекция №9. Использование массивов при программировании задач.

Массив - это набор данных одного типа. Массив записывается следующим образом: *тип_переменной имя_массива [длина_массива]*. После типа переменной отображается имя массива, затем его размеры отображаются в квадратных скобках. Например, мы определяем массив из 4 чисел: *int mas [4]*. Этот массив имеет четыре элемента, но их значения не определены. Мы можем дать этим элементам начальные значения в скобках: *int mas [4] = {1,2,3,4};*

Если число элементов меньше чем размера массива, то значения передаются начальным элементам массива. Если число элементов превышает размер массива, во время компиляции возникает ошибка: *int mas [4] = {1, 2, 3, 4, 5, 6};*. Здесь размер массива равно на 4, но ему присваивается 6 значений. Если размер массива не указан, то размер определяется числом указанных значений: *int mas [] = {1, 2, 3, 4, 5, 6};*. В этом случае массив содержит 6 элементов, поэтому размер массива равен 6.

Инициализация (присвоение) символьных массивов имеет свои особенности. Массив символьного типа можно дать с помощью символов, а так же в виде строк:

```
char s1 [] = {'h', 'e', 'l', 'l', 'o'};  
char s2 [] = "dunyo";
```

Кроме того, во втором случае массив s2 имеет 6 элементов вместо 5. Это потому, что когда вы присваиваете значение массиву строковым виде, в конец массива автоматически добавляется нулевой символ '\0'. Однако нельзя присваивать элементы одного массива другому массиву в виде:

```
int nums1 [] = {1,2,3,4,5};  
int nums2 [] = nums1; // неправильно  
nums2 = nums1; // неправильно
```

Определив массив, мы можем ссылаться на его элементы через индекс. Индексы начинаются с нуля, поэтому вы должны использовать индекс 0 для доступа к первому элементу. Обращаясь к элементу через индекс, мы можем получить или изменить его значение:

Пример №1.

```
#include <iostream>  
using namespace std;  
int main()  
{  
    int numbers[4] = {1,2,3,4};  
    int first_number = numbers[0];  
    cout << first_number<<endl; // 1
```

```

    numbers[0] = 34;
    cout << numbers[0] << endl; // 34
    return 0;
}

```

Используя цикл, через индексы можно ссылаться на элементы массива:

Пример №2.

```

#include <iostream>
using namespace std;
int main()
{
    int numbers[4] = {1,2,3,4};
    int size = sizeof(numbers)/sizeof(numbers[0]);
    for(int i=0; i < size; i++)
        cout << numbers[i] << endl;
    return 0;
}

```

Чтобы пройти массив через цикл, сначала должны найти длину массива. Оператор `sizeof` используется для определения длины. Фактически длина массива равна общей длине его элементов. Все элементы имеют одинаковый тип и занимают одинаковый размер в памяти. Таким образом, используя выражение `sizeof (numbers)`, мы находим всю длину массива в байтах, а используя `sizeof (numbers [0])`, мы находим длину одного элемента в байтах. Затем, используя оператор `for`, мы ссылаемся на все элементы, пока значение счетчика `i` не станет равным длине массива.

Пример №3. Вычислите среднее арифметическое всех положительных элементов вещественного массива a (n), $n \leq 15$.

Вводим обозначения. i - переменная, определяющая значения индексов элементов массива, $i = 1, 2, \dots, k$ (k - количество элементов массива, ($k \leq 15$), S и kn - среднее арифметическое положительных элементов и их количество соответственно. Если $a [i] > 0$, то $S = S + a [i]$, $kn = kn + 1$. Когда этот расчет завершен, вычисляется $S = S / kn$, т.е. сумма положительных элементов делится на количество положительных элементов. Если $kn = 0$, в массиве нет положительных элементов.

```

#include <iostream>
using namespace std;
int main()
{
    int i,k,kn;
    float s;
    cout<<"Ввести значение k \n"; cin>>k;
    float a[k]; s=0; kn=0;
}

```

```

    cout<<" Ввести массив a \n";
    for (i=1;i<=k; i++) cin>>a[i];
    for (i=1;i<=k; i++)
    if (a[i]>0)
    {s=s+a[i]; kn=kn+1;    }
    if (kn==0){
    cout<<"Не имеются положительных элементов \n";
    goto t;}
    else
    {s=s/kn;
    cout<<"s="<<s;
    }
    t:;}

```

Пример №4. Найдите наибольший элемент одномерного вещественного $b(n)$, $n = 20$ и его порядковый номер.

Определения: b_{max} и i_{max} - наибольший элемент массива $b(n)$ и его порядковый номер; Пусть k будет количеством элементов в массиве. Сначала мы вводим значение всех элементов массива b . Предположим, что первый элемент массива является самым большим элементом: $b_{max} = b_1$, $i_{max} = 1$. Параметр цикла i изменяется от 2 до k . Каждый раз, когда проверяется условие $b_i > b_{max}$, если выполняется это условие то пишем $b_{max} = b_i$ и $i_{max} = i$.

```

#include <iostream>
using namespace std;
int main()
{
    //    const int n=5;
    int i,k,imax;
    float bmax;
    //float b[n];
    cout<<"k ni kiriting"<<endl;
    cin>>k;
    float b[k];
    cout<<"b massivni kiriting \n";
    for (i=1;i<=k;i++)cin>>b[i];
    {bmax=b[1]; imax=1;}
    for (i=1;i<=k;i++)
    {
    if (b[i]>bmax)
    {bmax=b[i]; imax=i;}
    }
    cout<<"bmax="<<bmax<<"\t imax="<<imax;
}

```

Варианты самостоятельной работы:

Вариант №1

Найдите сумму элементов массива A (10).

Вариант №2

Найдите сумму элементов массива A (20) кратных на 3.

Вариант №3

Составить программу для вычисления среднего арифметического числа расположенных в нечетных местах элементов массива A (30).

Вариант №4

Найти сумму элементов расположенных в четных местах массива A (30), произведение отрицательных элементов на количество нечетных элементов.

Вариант №5

Найдите наибольший элемент массива B (30) и его порядковый номер.

Вариант №6

Найти среднее арифметическое элементов массива A (30) больше чем s.

Вариант №7

Вывести на экран нечетные элементы массива A (20).

Вариант №8

Запишите последовательность чисел C (30) в порядке убывания.

Вариант №9

Создайте массив В из положительных элементов массива А (20).

Вариант №10

Создайте массив В, состоящий из нечетных элементов массива А (20).

Вариант №11

Построить массив В, состоящий из положительных элементов массива А (20), и найти наибольший элемент массива В.

Вариант №12

Найдите наименьший из нечетных элементов массива А (20).

Вариант №13

Найдите наибольшее значение массива А (20) и его местоположение.

Вариант №14

Найдите наименьшее значение массива А (20) и его местоположение.

Вариант №15

Расположите элементы массива А (20) в обратном порядке.

Вариант №16

Найдите сумму и количество элементов массива А(40).

Вариант №17

Найти сумму элементов массива А (20) кратных на 3.

Вариант №18

Поместите в массив В (20) сначала положительные элементы, затем отрицательные элементы массива А (20).

Вариант №19

Вычислить среднюю геометрическую элементов массива А (20).

Вариант №20

Определите максимальный элемент массива А (20) и его местоположение.