

Лабораторная работа №4

Работа с массивами. Одномерные массивы

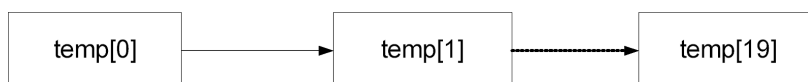
Цель работы: Ознакомиться со структурой массивов. Понять, как объявлять одномерный массив и обращаться к отдельным элементам массива.

Теоретические сведения

Массив - это тип данных, который используется для представления последовательности однородных значений. Массив представляет собой группу элементов одного типа. Объявляется массив следующим образом:

```
int temp[20];
```

Квадратные скобки ([]) говоря о том, что temp – имя массива, а число, заключенное в скобки, указывает количество элементов массива. Нумерация элементов массива начинается с нуля, поэтому temp[0] является первым, а temp[19] последним элементом массива. Отдельный элемент массива определяется при помощи его номера или индекса. Элементы массива размещаются в памяти последовательно, друг за другом. Имя массива является указателем на первый элемент массива.



```
float mas[10]; // массив mas содержит 10 элементов типа float
```

```
int mas1[20]; // массив mas1 содержит 20 элементов типа int
```

```
int n=5;
```

```
int mm[n]; // ошибка, переменная не может задавать размер массива
```

При объявлении массивы можно инициализировать. Для этого при объявлении указывается список начальных значений элементов, заключенных в фигурные скобки:

```
int mas[5]={1,2,3,4,5};
```

Количество элементов в фигурных скобках не должно превышать размерность массива. Инициализировать можно не все элементы, а любое количество первых элементов:

```
int mas[10]={1,2,3,4,5,6};
```

Все остальные четыре элемента будут проинициализированы нулями.

Инициализировать массив можно следующим образом:

```
int mas[]={11,22,33,44,55}; // массив из пяти элементов
```

В этом случае число элементов массива определяется по списку инициализации.

Пример: Ввести массив чисел и вычислить сумму положительных значений

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#define N 100
```

```

void main(void)
{
    int mas[N];           // массив чисел
    int summa=0;           // сумма положительных значений
    int n;                 // количество чисел
    int i;
    printf("\nВведите количество чисел не более %3d",N-1);
    scanf("%d",&n);
    printf("\nВведите %3d элементов массива",n);
    for(i=0; i < n; i++)    // цикл по элементам массива
        scanf("%d", &mas[i]); // ввод i-го элемента массива
    for(i=0; i < n; i++)
    {
        if(mas[i]>0)        // если i-ый элемент массива
положительный
            summa+=mas[i]; // суммируются положительные элементы
    }
    printf("\n сумма положительных значений равна %3d",summa);
}

```

Порядок выполнения работы

1. Изучить краткие теоретические сведения.
2. Составить блок-схему алгоритма.
3. По разработанной блок-схеме алгоритма написать программу.
4. Отладить и выполнить программу.

Варианты заданий

1. Дано натуральное n ($n \geq 2$). Найти все меньшие n простые числа используя решето Эратосфена. Решетом Эратосфена называют следующий способ. Выпишем подряд все целые числа от 2 до n . Первое простое число 2. Подчеркнем его, а все большие числа, кратные 2, зачеркнем. Первое из оставшихся чисел 3. Подчеркнем его как простое, а все большие числа, кратные 3, зачеркнем. Первое число из оставшихся теперь 5, так как 4 уже зачеркнуто. Подчеркнем его как простое, а все большие числа, кратные 5, зачеркнем и т.д.

2. Пусть дан массив a_1, \dots, a_n . Требуется переставить a_1, \dots, a_n так, чтобы вначале в массиве шла группа элементов, больших того элемента, который в исходном массиве располагался на первом месте, затем – сам этот элемент, потом – группа элементов, меньших или равных ему.

3. Пусть по некоторому каналу связи передается сообщение, имеющее вид последовательности нулей и единиц. Из-за помех возможен ошибочный прием некоторых сигналов: нуль может быть воспринят как единица и

наоборот. Можно передавать каждый сигнал трижды, заменяя, например, последовательность 1,0,1 последовательностью 1,1,1, 0,0,0, 1,1,1. Три последовательные цифры при расшифровке заменяются той цифрой, которая встречается среди них по крайней мере дважды. Такое утраивание сигналов существенно повышает вероятность правильного приема сообщения. Написать программу расшифровки.

4. Задан массив положительных чисел a_1, \dots, a_n . Для каждого $a[i]$ среди элементов массива следующих по порядку за $a[i]$ и больших чем $a[i]$, выберем элемент с наименьшим номером j и заменим значение $a[i]$ на $a[j]$. Если такого элемента $a[j]$ не найдется, то заменим значение $a[i]$ нулем. Распечатать получившийся массив.

Пояснение. Например, массив 2,9,8,5,9,3,4,5,2 после замены станет таким 9,0,9,9,0,4,5,0,0.

5. Дан массив a_1, \dots, a_n . Исключить из него пять минимальных элементов, сдвинув оставшиеся элементы к левому краю.

6. Даны два целочисленных массива a_1, \dots, a_n и b_1, \dots, b_n . Вывести на печать все пары индексов, для которых $a[i] * b[i] > 10$. Подсчитать число пар и сумму этих произведений.

7. Дан одномерный массив a_1, \dots, a_n . Найти и напечатать номер элемента, произведение которого с предыдущим максимально.

8. Рассортировать одномерный массив по возрастанию (убыванию) элементов (метод Шелла).

9. Рассортировать одномерный массив по возрастанию (убыванию) элементов (метод пузырька).

10. В одномерном массиве a_1, \dots, a_n заменить отрицательные элементы нулями, подсчитать число замен m , вычислить $m!$.

11. Ввести целочисленный массив, состоящий из 15 элементов.

Определить сумму и разность максимального и минимального элементов

12. Дан массив $M(15)$ вещественных чисел. Расположить элементы в обратном порядке.

13. В заданном целочисленном массиве $R(9)$ определить индекс наибольшего из нечетных по значению положительных элементов.

14. Ввести массив, состоящий из 10-ти элементов целого типа.

Сформировать новый, расположив сначала все отрицательные элементы и нули, после чего - положительные, сохраняя порядок их следования.

15. Задан целочисленный массив. Определить количество участков массива, на котором элементы монотонно возрастают (каждое следующее число больше предыдущего).