

## ЛАБОРАТОРНАЯ РАБОТА №5

### МАССИВЫ И УКАЗАТЕЛИ

**Цель работы:** изучить правила работы с массивами, связь массивов с указателями.

#### Краткие теоретические сведения

#### ОДНОМЕРНЫЙ МАССИВ

**Массив** – это группа расположенных друг за другом элементов в памяти переменных **одного типа** и имеющих одно **общее имя**. Каждому элементу массива отводится одна ячейка памяти. Элементы одного массива занимают последовательно расположенные ячейки памяти.

#### *Определение одномерного массива*

**ТипДанных ИмяМассива[Размер];**

**ТипДанных** – это любой тип, определенный в программе, **кроме функции** и типа *void*.

**ИмяМассива** – это **скрытый константный указатель** на первый элемент массива.

**Размер** для любого типа массива – это **целая положительная константа**.

#### *Пример*

```
const int N = 6; //определение размера массива
```

```
double arr[N]; //определение массива вещественных чисел
```

**arr** (имя массива) – это адрес первого элемента массива, т.е. **arr** – это **&arr[0]**

Чтобы обратиться к элементу массива, надо указать имя массива и номер элемента в массиве (индекс): **arr[ind]** – это число вещественного типа, где **индекс** – это **целое положительное число**, которое может меняться в интервале **[0, N-1]**, где **N** – **размер**.

При обработке одномерного массива используется цикл, например, ввод массива.

```

int ind;//определение индекса массива
for(ind = 0; ind < N; ind++){
    printf("[%d]=",ind);//выводим номер ячейки, куда вводится значение
    scanf("%lf", &arr[ind]);// ввод значения в ind-й номер ячейки
}

```

## СВЯЗЬ УКАЗАТЕЛЕЙ И МАССИВОВ

```

const int N = 6;

double arr[N]={23.1, -34, 90.7, 34,-45, 10.8};

double *p;

```

Направить указатель на массив можно двумя способами:

1. `p = arr;`
2. `p = &arr[0];`

Обращения к элементу массива **arr**: `arr[i]` , `*(arr+i)` и `*(p+i)` эквивалентны.

Таким образом, для любых указателей можно использовать две эквивалентные формы выражений для доступа к элементам массива: `arr[i]` и `*(arr+i)`. Первая форма удобнее для читаемости текста, вторая – эффективнее по быстродействию программы.

Ввод элементов массива через указатели:

```

int ind;//определение индекса массива
for(ind = 0; ind < N; ind++)
{
    printf("[%d]=",ind);//выводим номер ячейки, куда вводится значение
    scanf("%lf", (arr+ind));// ввод значения в ind-й номер ячейки
}

```

## ПЕРЕДАЧА ОДНОМЕРНОГО МАССИВА В ФУНКЦИЮ

**ТипВозврЗн ИмяФункции (ТипДанных \*, int);**

При вызове функции параметры необходимо передавать следующим образом:

- первый параметр – это имя массива;
- второй параметр – это размер массива.

## МНОГОМЕРНЫЕ МАССИВЫ

**Многомерный массив** – это массив, элементами которого служат массивы.

### *Определение двумерного массива*

**ТипДанных ИмяМассива[размер1][размер2];**

**Размер1**– это **целая положительная константа**, отражающая **количество строк**, т.е количество указателей, которые содержат имена одномерных массивов.

**Размер2**– это **целая положительная константа**, отражающая **количество столбцов**, т.е. размер одномерного массива.

**ИмяМассива** – это **адрес адреса**, т.е. указатель на указатель.

По определению многомерные массивы в C/C++ не существуют. Если мы описываем массив с несколькими индексами, например, массив **int mas[3][4]**, то это означает, что мы описали одномерный массив **mas**, элементами которого являются 3 указателя на одномерные массивы, размер каждого целочисленного массива 4.

Доступ к элементам многомерных массивов возможен и с помощью индексированных переменных и с помощью указателей:

**mas[2][1]** – доступ с помощью индексированных переменных,

**\*(\*(mas+2)+1)** – доступ к этому же элементу с помощью указателей.

При обработке двумерных массивов используются вложенные циклы, например, ввод массива **int mas[5][4]**:

```
for(i=0; i<5; i++)
```

```
    for(j=0; j<4; j++)
```

```
        scanf("%d", &mas[i][j]);
```

## ПЕРЕДАЧА МНОГОМЕРНОГО МАССИВА В ФУНКЦИЮ:

ТипВозврЗн ИмяФункции (ТипДанных [ ][ размер2], int);

При вызове функции параметры необходимо передавать следующим образом:

- первый параметр – это имя массива и обязательно размер одномерного массива;
- второй параметр – это количество строк массива, т.е. количество указателей.

## ПРИМЕРЫ РЕШЕНИЙ

### 1. Нахождение максимального элемента массива вещественных чисел и его место.

```
#include<stdio.h>

void main() {

    const int n=6;

    double x[n];

    int i, i_max=0;

    for(i=0;i<n;i++) {

        printf("x[%d]=", i);

        scanf("%lf",&x[i]);

        if(x[i]>x[i_max])

            i_max=i;

    }

    printf("max=x[%d]=%.2lf\n", i_max, x[i_max]);

}
```

Рассмотрим данный алгоритм подробнее. **const int n=6;** – это размерность массива, который задается с помощью константы **n**.

**double** x[n]; -- это объявление массива вещественных чисел. **int** i, i\_max=0; переменная **i** для прохода по циклу и отображает индекс массива, переменная **i\_max** для хранения индекса максимального элемента. Мы предполагаем, что первый элемент массива, т.е. **x[0]**, и будет максимальным, следовательно, и начальное значение переменной **i\_max** и есть 0. Далее мы с помощью оператора **for** вводим данные в массив и одновременно сравниваем с предполагаемым максимумом. После выхода из цикла в переменной **i\_max** будет храниться индекс максимального элемента массива, а к самому элементу можно будет обратиться следующим образом **x[i\_max]**

## 2. Ввести двухмерный массив и отсортировать его.

```
#include<stdio.h>

void main() {

    const int n=4,m=3;

    int mat[n][m],i,j,temp;

    for(i=0;i<n;i++){

        printf("\nВвод %d-ой строки: ",i+1);

        for(j=0;j<m;j++)

            scanf("%d",&mat[i][j]);

    }

    for(i=0;i<n;i++){

        printf("\n\n");

        for(j=0;j<m;j++)

            printf("%d ",mat[i][j]);

    }

    for(i=0;i<n*m;i++){

        for(j=0;j<m*n;j++){

            if(mat[i/m][i%m]<mat[j/m][j%m]){

                temp=mat[i/m][i%m];

                mat[i/m][i%m]=mat[j/m][j%m];
```

```

        mat[j/m][j%m]=temp;

    }

}

for(i=0;i<n;i++){
    printf("\n\n");
    for(j=0;j<m;j++)
        printf("%d ",mat[i][j]);
}
}

```

### **ЗАДАЧИ ДЛЯ ВЫПОЛНЕНИЯ:**

1. Посчитать сумму элементов массива. Размер массива 7, данные вводятся с клавиатуры.
2. Программа запрашивает с клавиатуры десять вещественных чисел, а затем выводит их в обратном порядке.
3. Посчитать сумму элементов массива целых чисел с четным индексом.
4. Ввести массив вещественных чисел. Посчитать: а) количество отрицательных элементов массива; б) сумма положительных элементов массива; в) произведение элементов, находящихся под нечетным индексом.
5. Посчитать, сколько пятерок, четверок, троек и двоек за контрольную в классе из 20 человек. (Для заполнения массива использовать генерацию случайных чисел).
6. Задан массив размерностью 10. Ввести число, которое необходимо записать под 5-м индексом, причем следующим образом введенное число записывается на 5-ом месте, число записанное на 5-м месте смещается на 6-е, 6-е на 7-е и т.д., впоследствии последнее число, которое было записано в массиве теряется. (т.е. сдвиг массива на один шаг вправо).

7. Ввести массив. Размер массива 20. Найти максимальный элемент в первой половине массива и минимальный во второй половине массива. Поменять их местами. Вывести содержание измененного массива.
8. Введите два индекса и поменяйте местами элементы, находящиеся под данными индексами. Выведите содержание измененного массива.
9. При определении массива вещественных чисел, размером 20, сразу заполнить его. Отсортировать массив по убыванию методом Шелла. Вывести отсортированный массив. В программе написать функции вывода и сортировки.
10. Ввести данные в массив размером 20. Отсортировать первую половину массива методом вставки, по возрастанию, и вторую половину пузырьковым методом, по убыванию. Вывести отсортированный массив. В программе написать функции вывода, ввода и сортировок.
11. Даны координаты  $n$  точек.  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ . Найти координаты точек, которые находятся на максимальном расстоянии друг от друга.
12. Написать следующие функции для одномерного массива: 1. ввод массива; 2. вывод массива; 3. нахождение и максимального элемента и его места; 4. нахождение и минимального элемента и его места; 5. сортировка массива по убыванию; 6. сортировка массива по возрастанию.

В программе использовать меню (указатель на функции). Программа завершает работу при вводе 0.

13. Ввести двумерный массив. Посчитать сумму элементов находящихся на диагонали, место максимального элемента выше главной диагонали и место минимального элемента ниже главной диагонали.

14. Ввести двумерный массив. Отсортировать его по столбцам.
15. Перемножение матриц. Использовать функции ввода и вывода массивов.
16. Дана целочисленная матрица. Посчитать сумму элементов тех строк, в которых не содержится 0.

17. Дана матрица. Посчитать произведение тех элементов, которые находятся между первым и последним отрицательным элементами.
18. Дана целочисленная прямоугольная матрица. Определить номер строк и столбцов всех седловых точек матрицы. Матрица имеет седловую точку  $x[i][j]$ , если  $x[i][j]$  является минимальным элементом в  $i$ -й строке и максимальным в  $j$ -м столбце.