

## Тема 5. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ ОДНОМЕРНЫХ МАССИВОВ

**Цель занятия:** получить практические навыки в работе с массивами; знакомство с задачами, для решения которых используются массивы.

### Теоретические сведения

**Массив (array)** - это конечный набор элементов одного (базового) типа, которые сохраняются в последовательно размещённых ячейках оперативной памяти и имеют общее имя.

В математике понятию массив соответствуют понятия вектора и матрицы. Различают одно- и многомерные массивы. Двумерный массив данных – это таблица, которая состоит из нескольких строк.

*Для заданного типа type объявление массива:*

**type array\_name[size];**

формирует в памяти массив из *size* элементов типа *type*.

Элементы индексируются **от 0 до size-1**, т.е.

*v[10]* – это *v[0]...[9]*.

*int a[100];* /массив из 100 элементов целого типа

Операция *sizeof(a)* даст результат 400, т.е. 100 элементов по 4 байта.

*Для обращения к значению элемента массива* задаётся имя массива (*array\_name*) и индекс элемента массива (*index*) в квадратных скобках:

**array\_name[index]**

*a[0]* – индекс задается как константа,

*a[I]* – индекс задается как переменная,

*a[2\*I]* – индекс задается как выражение.

Элементы массива можно задавать при его определении:

*int v1[4] = {1, 2, 3, 4};*

*char v2[4] = {'a', 'b', 'c', 'd'};*

Когда массив объявлен без указания размера, но при этом инициализируется списком, размер массива вычисляется путём подсчёта числа элементов этого списка:

*int v3[] = {1, 2, 3, 4};*

*char v4[] = {'a', 'b', 'c', 'd'};*

Здесь *v3* и *v4* - массивы из четырех символов. Длина массива вычисляется компилятором по количеству значений, перечисленных при инициализации.

Если количество начальных значений меньше, чем объявленная длина массива, то начальные элементы массива получают только первые элементы:

*int a[12]={1,2,3,4,5,6,7,8,9,10};*

*Если в списке инициализации недостаёт элементов, всем остальным элементам присваиваются нулевые значения.* Например:

*int v5[8] = {1, 2, 3, 4};* равнозначно *int v5[8] = {1, 2, 3, 4, 0, 0, 0, 0};*

*Если размер задан явно, присваивать большее число элементов нельзя.*

Массив символов можно задавать в виде строки инициализатора:

*char alpha[] = "abcdefghijklmnopqrstuvwxyz";*

Не нужно указывать размер массива *alpha*: компилятор установит его, подсчитав число символов в строке, заданной в качестве инициализатора.

Присваивание строки массиву недопустимо, поскольку в языке присваивание массивам не определено, например:

```
char v[9];  
v = "a string";    // ошибка!
```

Многомерные массивы представляются как массивы массивов.

```
int d2[10][20]; // d2 является массивом из 10 элементов по 20 элементов каждый  
char v[2][5] = { {'a', 'b', 'c', 'd', 'e'}, {'0', '1', '2', '3', '4'} };
```

**Пример.** Ввод-вывод массива

```
int main()  
{  
    SetConsoleOutputCP(1251);  
    const int k = 20;  
    int a[k];  
    //Ввод массива  
    cout<<"Введите элементы массива :\n";  
    for(int i = 0; i < k; i++)  
        cin>>a[i];  
    //Вывод массива  
    cout<<"Массив:\n";  
    for(int i = 0; i < k; i++)  
        cout<<a[i]<<" ";  
}
```

В C++ имеется специальная функция *int rand()* – возвращает псевдослучайное число из диапазона 0 ... RAND\_MAX=32767, описание функции находится в файле *<stdlib.h>*.

**Пример.** Ввод массива с использованием датчика случайных чисел

```
int main()  
{  
    SetConsoleOutputCP(1251);  
    const int k = 20;  
    int a[k];  
    //Ввод массива  
    cout<<"Введите элементы массива :\n";  
    for(int i = 0; i < k; i++)  
        a[i] = rand()% 100-50;  
}
```

### Перебор массива по одному элементу

Элементы можно перебирать:

1. Слева направо с шагом 1, используя цикл с параметром:  
*for (int I=0; I<n; I++)* { обработка a[I]; }
2. Слева направо с шагом отличным от 1, используя цикл с параметром:  
*for (int I=0; I<n; I+=step)* { обработка a[I]; }
3. Справа налево с шагом 1, используя цикл с параметром:  
*for (int I=n-1; I>=0; I--)* { обработка a[I]; }
4. Справа налево с шагом отличным от 1, используя цикл с параметром:

*for (int I=n-1;I>=0;I-=step) {обработка a[I];}*

### **Перебор массива по два элемента**

1) Элементы массива можно обрабатывать по два элемента, двигаясь с обеих сторон массива к его середине:

```
int I=0, J=N-1;
while( I<J)
{обработка a[I] и a[J];
I++;J--;}
```

2) Элементы массива можно обрабатывать по два элемента, двигаясь от начала к концу с шагом 1 (т.е. обрабатываются пары элементов a[1] и a[2], a[2] и a[3] и т.д.):

```
for (I=0;I<N-1;I++)
{обработка a[I] и a[I+1]}
```

3) Элементы массива можно обрабатывать по два элемента, двигаясь от начала к концу с шагом 2 (т.е. обрабатываются пары элементов a[1] и a[2], a[3] и a[4] и т.д.)

```
int I=0;
while (I<N-1 )
{обработка a[I] и a[I+1];
I+=2;}
```

### **Классы задач по обработке массивов**

1. К задачам 1 класса относятся задачи, в которых выполняется однотипная обработка всех или указанных элементов массива.

2. К задачам 2 класса относятся задачи, в которых изменяется порядок следования элементов массива.

3. К задачам 3 класса относятся задачи, в которых выполняется обработка нескольких массивов или подмассивов одного массива. Массивы могут обрабатываться по одной схеме – синхронная обработка или по разным схемам – асинхронная обработка массивов.

4. К задачам 4 класса относятся задачи, в которых требуется отыскать первый элемент массива, совпадающий с заданным значением – поисковые задачи в массиве.

### **Задачи 1-го класса**

Решение таких задач сводится к установлению того, как обрабатывается каждый элемент массива или указанные элементы, затем подбирается подходящая схема перебора, в которую вставляются операторы обработки элементов массива.

**Пример 5.1.** Нахождение среднего арифметического массива.

```
#include<iostream>
#include<stdlib.h>
#include <locale>
using namespace std;
void main()
{
    setlocale (LC_CTYPE,"rus");
    int a[100];
    int n,i;
    cout<<"\nEnter the size of array:";
    cin>>n;
```

```

for( I=0; I<n; I++)
{
    a[I]=rand()%100-50;
    cout<<a[I]<<" ";
}
double Sum=0;
for(I=0; I<n; I++)
    Sum+=a[I];
cout<<"Среднее арифметическое="<<Sum/n;
}

```

**Задание 5.1.** Найти среднее арифметическое отрицательных элементов массива.

**Пример 5.2.** Нахождение максимального элемента массива и его позиции.

```

#include<iostream>
#include<stdlib.h>
#include <locale>
using namespace std;
void main()
{
    setlocale (LC_CTYPE,"rus");
    int a[100];
    int n,I;
    cout<<"\nEnter the size of array:";
    cin>>n;
    for( I=0; I<n; I++)
    {
        a[I]=rand()%100-50;
        cout<<a[I]<<" ";
    }
    int max=a[0], nMax=0;
    for(I=1;I<n;I++)
        if (a[I]>max) {max=a[I];nMax=I;}
    cout<<"\n Max="<<max<<" его номер в массиве "<<nMax+1;
}

```

**Задание 5.2.** Найти минимальный элемента и его позицию в массиве.

**Пример 5.3.** Найти сумму элементов массива с четными индексами.

```

//формирование массива
int Sum=0;
for(I=0; I<n; I+=2)
    Sum+=a[I]; //элементы с индексами 0, 2, 4...
cout<<"\nSum="<<Sum;

//Второй способ
int Sum=0;
for (I=0; I<n; I++)
    if(I%2==0) Sum+=a[I]; //элементы с индексами 0, 2, 4...
cout<<"\nSum="<<Sum;

```

**Задание 5.3.** Написать и отладить программу по условию Примера 5.3

#### Задачи 2-го класса

Обмен элементов внутри массива выполняется с использованием вспомогательной переменной:

```
int R=a[I];a[I]=a[J]; a[J]=R; // обмен a[I] и a[J] элементов массива.
```

**Пример 5.4.** Перевернуть массив.

```
//формирование массива
for (int i=0, j=n-1; i<j; i++, j--)
{int r=a[i];
a[i]=a[j];
a[j]=r;}
//Вывод массива
```

**Пример 5.5.** Поменять местами пары элементов в массиве: 1 и 2, 3 и 4, 5 и 6 и т.д.

```
//формирование массива
for (int i=0; i<n-1; i+=2)
{int r=a[i];
a[i]=a[i+1];
a[i+1]=r;}
//Вывод массива
```

**Пример 5.6.** Циклически сдвинуть массив на k элементов влево.

```
//формирование массива
int k,i,t,r;
cout<<"\nK=?";
cin>>k;
for (t=0; t<k; t++)
{
r=a[0];
for (int i=0; i<n-1; i++)
a[i]=a[i+1];
a[n-1]=r;
}
//Вывод массива
```

**Задание 5.4.** Циклически сдвинуть массив на k элементов вправо.

#### Задачи 3-го класса

При синхронной обработке массивов индексы при переборе массивов меняются одинаково.

**Пример 5.7.** Заданы два массива из n целых элементов. Получить массив c, где  $c[i]=a[i]+b[i]$ .

```
//формирование массива a и b
for (int i=0; i<n; i++) c[i]=a[i]+b[i];
//Вывод массива
```

**Задание 5.5.** Написать и отладить программу по условию **Примера 5.7**

При асинхронной обработке массивов индекс каждого массива меняется по своей схеме.

**Пример 5.8** В массиве целых чисел все отрицательные элементы перенести в начало массива.

```
//формирование массива a
int b[12];           //вспомогательный массив
int i, j=0;
for (i=0; i<n; i++)
if (a[i]<0){b[j]=a[i]; j++;} //переписываем из a в b все отрицательные элементы
for (i=0; i<n; i++)
if (a[i]>=0){b[j]=a[i]; j++;} // переписываем из a в b все положительные элементы
for (i=0; i<j; i++) cout<<b[i]<< " "; //вывод массива b
```

**Задание 5.6.** В массиве вещественных чисел все ненулевые элементы перенести в конец массива.

**Пример 5.9.** Удалить из массива все четные числа

```
//формирование массива a
int b[12];
int i, j=0;
for (i=0; i<n; i++)
if (a[i]%2!=0) {b[j]=a[i]; j++;}
for i=0; i<j; i++) cout<<b[i]<< " "; //переписываем из a в b все нечетные элементы
cout<<"\n";
```

**Задание 5.6.** Удалить из символьного массива все символы '\*'

#### Задачи 4-го класса

В поисковых задачах требуется найти элемент, удовлетворяющий заданному условию. Для этого требуется организовать перебор массива и проверку условия. Но при этом существует две возможности выхода из цикла:

- 1) нужный элемент найден;
- 2) элемент не найден, но просмотр массива закончен.

**Пример 5.10.** Найти первое вхождение элемента K в массив целых чисел.

```
//формирование массива a
int k;
cout<<"\nK=?";
cin>>k;
int ok=0;           //признак найден элемент или нет
int i,nom;
for(i=0; i<n; i++)
if(a[i]==k){ok=1; nom=i; break;}
if(ok==1)
```

```
cout<<"\nnom="<<nom;
else
cout<<"\nthere is no such element! ";
```

**Задание 5.7.** Найти последнее вхождение заданного символа в массив символов.

### Методические указания

При подготовке к занятию необходимо изучить: способы описания массивов на языке C++; особенности ввода и вывода массивов; типичные операции над массивами.

### Аудиторные и домашние задания

- Даны натуральные числа  $n_1, \dots, n_{20}$ , действительные числа  $x_1, \dots, x_{20}$ . Вычислить  $\frac{n_1 x_1 + \dots + n_{20} x_{20}}{n_1 + \dots + n_{20}}$ .
- Даны целые числа  $n, a_1, \dots, a_n$ . Получить новую последовательность, удалив из исходной все нечетные числа.
- Дана последовательность из 100 различных целых чисел. Найти сумму чисел этой последовательности, расположенных между максимальным и минимальными числами (в сумму включить и оба этих числа).
- Даны натуральное число  $n$ , целые числа  $a_1, \dots, a_n$ . Все члены последовательности  $a_1, \dots, a_n$ , предшествующие первому по порядку наименьшему члену, домножить на этот наименьший член.
- Даны натуральные числа  $n, a_1, \dots, a_n$ . Определить количество членов  $a_k$  последовательности  $a_1, \dots, a_n$ , удовлетворяющих условию  $a_k < \frac{a_{k-1} + a_{k+1}}{2}$ .
- Даны натуральное число  $n$ , символы  $s_1, \dots, s_n$ . Преобразовать последовательность  $s_1, \dots, s_n$ : если нет символа \*, то оставить ее без изменения, иначе заменить каждый символ, встречающийся после первого вхождения символа \*, на символ -.
- Даны натуральное число  $n$ , действительные числа  $x_1, \dots, x_n$ . Вычислить  $(x_1 + x_2 + 2x_n)(x_2 + x_3 + 2x_{n-1})(x_{n-1} + x_n + 2x_2)$ .
- Даны натуральное число  $n$ , действительные числа  $x_1, \dots, x_{3n}$ . Вычислить сумму чисел из  $x_{n+1}, \dots, x_{3n}$ , которые превосходят по величине все числа  $x_1, \dots, x_n$ .
- Даны целые  $n, a_1, \dots, a_n$ . Из модулей членов данной последовательности выбрать наибольший. Получить новую последовательность из  $n$  целых чисел (нулей и единиц), заменяя  $a_i$  нулем, если  $|a_i|$  не совпадает с наибольшим значением, и заменяя  $a_i$  единицей, если совпадает.
- Дано натуральное число  $n$ , действительные числа  $a_1, \dots, a_n$ . Выполнить циклический сдвиг массива на  $k$  позиций вправо.
- Ввести массив целых чисел. Заменить последний элемент массива целой частью среднего арифметического этого массива.
- Дан текст, содержащий от 1 до 50 символов. Признак окончания ввода - точка.
  - Напечатать этот текст в обратном порядке.
  - Проверить, симметричен ли этот текст относительно середины, т. е. читается ли он одинаково слева направо и справа налево.
- Даны координаты  $n$  точек на плоскости:  $x_1, y_1, \dots, x_n, y_n$  ( $n=20$ ). Найти номера двух точек, расстояние между которыми наибольшее.
- Переменной  $k$  присвоить либо номер первого вхождения числа  $y$  в массив  $x(n)$ , либо число  $n+1$ , если  $y$  не входит в массив.

15. Дан массив целых чисел. Если в последовательности ни одно четное число не расположено после нечетного, то получить все отрицательные члены последовательности, иначе – положительные. Порядок следования чисел в обоих случаях заменяется на обратный.
16. Задан массив  $x(m)$ . Найти длину  $k$  самой длинной «пилообразной (зубьями вверх)» последовательности идущих подряд чисел:  $x_{p+1} < x_{p+2} > x_{p+3} < \dots > x_{p+k}$ .
17. В массиве  $a(k)$  определить количество различных чисел. Вывести эти числа.

### **Контрольные вопросы**

1. Что такое массив? Назовите его основные атрибуты.
2. Какие данные могут выступать в качестве индексов и элементов массива?
3. Как описываются массивы?
4. Укажите особенности ввода и вывода массивов.
5. В чем состоит особенность организации цикла при обработке массива?
6. Поясните назначение и правила использования операторов *break* и *continue*.