

## **Контрольная работа – 2 семестр.**

### **Одномерные массивы.**

#### **1. Цель лабораторной работы.**

- 1.1. Целью лабораторной работы является знакомство со структурными типами данных и получение практических навыков работы с одномерными массивами.
- 1.2. В результате выполнения лабораторной работы студенты должны знать:
- способы объявления массивов;
  - ввод и вывод массивов;
  - обращение к элементу массива и действия над ним.

**2. Теоретический материал для домашнего изучения.** С понятием «массив» приходится сталкиваться при решении задач обработки совокупностей большого количества значений. В общем случае **массив** – это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип.

Элементами массива могут быть данные любого типа, включая структурированные. Тип элементов массива называется базовым. Элементы, образующие массив, упорядочены таким образом, что каждому элементу соответствует совокупность номеров (**индексов**), определяющих его местоположение в общей последовательности. Доступ к каждому отдельному элементу осуществляется путем индексирования элементов массива. Элементы массива нумеруются индексами от 0 до  $n-1$ , где  $n$  - число элементов массива.

#### **Описание массивов**

Любая информация в ЭВМ хранится в ячейках оперативной памяти. Поэтому предполагаемая работа с массивом требует предварительного выделения памяти для его хранения. Это действие выполняет описание массива.

При описании массива указывается тип и число его элементов. Тип записывается перед именем массива, размер массива указывается в квадратных скобках после его имени. **Примеры:**

**int** a[10];      описан массив из 10 целых чисел (индекс меняется от 0 до 9)  
**double** r[1000];      описан вещ. массив из 1000 элементов

### Обозначение элементов массива

Для доступа к элементу массива указывается его имя и в квадратных скобках - индекс нужного элемента. С элементом массива можно работать как с обычной переменной, т.е. можно прочитать его значение или записать в него новое значение.

### Примеры:

a[3] = 0; элементу массива a с индексом 3 присваивается значение 0; a[10] = a[10]\*2; элемент массива a с индексом 10 удваивается.

**Инициализация** (присваивание начальных значений) массива заключается в присваивании каждому элементу массива одного и того же значения, соответствующего базовому типу. Наиболее эффективно эта операция выполняется с помощью оператора for, например:

```
for (i=0; i<10;i++) a[i]=0;
```

Инициализация может быть проведена при объявлении массива: **int**

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

### Ввод массивов

Ввод массива выполняется поэлементно аналогично вариантам ввода простых переменных.

Фрагмент программы ввода элементов массива имеет вид:

```
float a[20]; int i,n;  
cout << "Enter n (<20): "; cin >> n;  
for ( i=0 ; i<n ; i++ ){ cout << "Enter a [ "<< i << "  
]: "; cin >> a[i];  
}
```

Оператор **for** формирует цикл, запрашивая на каждом шаге текущее значение a(i+1) и организуя его ввод в ячейку элемента массива a[i]. **Вывод массивов.**

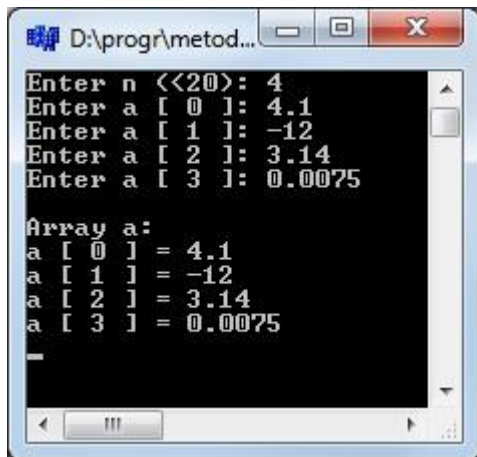
Вывод массива выполняется поэлементно аналогично вариантам вывода простых переменных.

Фрагмент программы вывода элементов массива имеет вид:

```
cout << "Array a:" << endl;
```

```
for( i=0 ; i<n ; i++ ) cout << "a [ " << i << " ] = " << a[i] << endl;
```

Реализация двух фрагментов программы выглядит следующим образом:



**Пример 1:** Найти наименьший элемент в массиве и его порядковый номер. Найти сумму первых 10 элементов и количество нечетных среди них.

```
int a[50], sum, k;      // массив
int min;               // номер минимального элемента int i,
n;                     // индекс массива и число элементов
cout << "Enter n (<50): ";
cin >> n; randomize();
cout<<"Array a:"<<endl;
for ( i = 0 ; i < n ; i++ ){ // заполняем массив случ. числами a[i]=random(100);
cout << "a [ " << i << " ] = " << a[i] << endl;
}
min =0; // предположим, что первый элемент - минимальный for ( i = 1; i < n;
i++) // сравним все оставшиеся элементы с минимальным
if (a[i] < a[min]) min = i;
cout << "Min element : \na [ " << min << " ] = " << a[min]; k=sum=0;
if ( n > 10 ){ for ( i = 0; i < 10;
i++ ){ sum=sum + a[i]; if ( a[i]
% 2 ) k ++;
}
cout << "Sum = " << sum << endl; cout
<< "k = " << k << endl; }
```

### **3. Выполнение задания.**

#### **3.1. Изучите и выполните примеры.**

#### **3.2. Составьте алгоритм и напишите программу по варианту.**

1. Дан одномерный массив. Определить сумму элементов массива с  $k_1$  по  $k_2$  (значения  $k_1$  и  $k_2$  вводятся с клавиатуры,  $k_2 > k_1$ ).
2. Составить программу, которая формирует одномерный массив вводом с клавиатуры и находит среднее арифметическое элементов массива с  $p_1$  по  $p_2$  (значения  $p_1$  и  $p_2$  вводятся с клавиатуры;  $p_2 > p_1$ ).
3. Дан одномерный массив  $a$ . Определить сумму знаочередующейся последовательности  $a[0]-a[1]+a[2]-a[3]+\dots$ . Условную инструкцию и операцию возведения в степень не использовать.
4. В массиве  $a[n]$  целых чисел найти сумму тех элементов, которые кратны данному значению  $k$ .
5. Дан массив целых чисел. Заменить все элементы, большие данного числа  $z$ , этим числом. Подсчитать количество замен.
6. Дан массив целых чисел. Подсчитать, сколько в нем отрицательных, положительных и нулевых элементов.
7. Дан массив целых чисел. Найти максимальный и минимальный элементы в массиве и поменять их местами.
8. Дан массив целых чисел. Вывести на печать только те числа, для которых выполняется условие  $a_i > i$ .
9. Дан массив целых чисел. Вывести на экран те из них, у которых остаток от деления на  $m$  равен  $f$  ( $0 \leq f \leq m-1$ ), и подсчитать их количество.
10. Дан массив целых чисел, среди которых имеется один нуль. Вывести на экран все числа до нуля включительно.
11. Дан массив целых чисел. Напечатать те его элементы, индексы которых являются степенями двойки (1, 2, 4, 8, 16, ...).
12. Дан массив целых чисел. Определить, сколько среди них чисел, меньших  $k$ , равных  $k$  и больших  $k$ .

#### **3.3. Составьте алгоритм и напишите программу по варианту.**

1. Вставить после первого положительного элемента в массиве два элемента, равных заданному значению. Если вставка элементов невозможна, выдать об этом сообщение.
2. Вставить после последнего отрицательного элемента в массиве два элемента, равных заданному значению. Если вставка элементов невозможна, выдать об этом сообщение.
3. Вставить после первого нулевого элемента в массиве два элемента, равных заданному значению. Если вставка элементов невозможна, выдать об этом сообщение.
4. Вставить заданное значение перед каждым элементом массива, расположенным после первого нулевого элемента. Если вставка элементов невозможна, выдать об этом сообщение.
5. Вставить в массив элемент с заданным значением после первого элемента с максимальным значением и перед первым элементом элемента с минимальным значением. Если вставка элементов невозможна, выдать об этом сообщение.
6. Вставить в массив элемент с заданным значением перед первым элементом с максимальным значением и после первого элемента с минимальным значением. Если вставка элементов невозможна, выдать об этом сообщение.
7. За каждым элементом с заданным значением вставить его дубликат. Если вставка элементов невозможна, выдать об этом сообщение.
8. Перед каждым элементом с отрицательным значением вставить элемент, равный абсолютной величине отрицательного элемента. Если вставка элементов невозможна, выдать об этом сообщение.
9. Удалить в массиве первую группу из двух подряд идущих нулей. Если удаление элементов невозможно, выдать об этом сообщение.
10. Удалить в массиве последнюю группу из двух подряд идущих положительных чисел. Если удаление элементов невозможно, выдать об этом сообщение.
11. Удалить в массиве первую группу из двух подряд идущих отрицательных чисел. Если удаление элементов невозможно, выдать об этом сообщение.

12. Удалить в массиве последнюю группу из двух подряд идущих одинаковых чисел. Если удаление элементов невозможно, выдать об этом сообщение.

Отчет: загрузите файлы с программами в виде архива и приложите краткий отчет со скриншотами работы программ.

