

ЛАБОРАТОРНАЯ РАБОТА № 13

ОДНОМЕРНЫЕ МАССИВЫ

Цель работы: получение практических навыков в работе с одномерными массивами.

Краткие теоретические сведения

Массив – это конечная именованная последовательность однотипных величин. Массивы широко используются в том случае, если необходимо обработать много данных, которые пользователь вводит с клавиатуры.

Массив называется *одномерным*, если он состоит из одной строки и N столбцов (вектор).

Формат описания массива следующий:

тип_элементов имя [размерность];

Размерность – это количество элементов в массиве (целая положительная константа).

Например:

```
int array[10];
```

Данная запись означает, что резервируется память для 10 чисел целого типа с именем `array` и порядковыми номерами (индексами) от 0 до 9 включительно. **В Си индексация начинается с нуля**, поэтому размер массива, указанный в описании, всегда на единицу больше максимального значения индекса.

Чтобы обратиться к отдельному элементу массива, надо указать имя массива и номер элемента в массиве (индекс) в квадратных скобках. Индекс является целым числом.

`array[0]` – первый элемент массива, `array[9]` – последний элемент массива.

Элементы массива обладают всеми атрибутами заданного типа.

Существуют два варианта объявления массива:

- **неинициализированный** (например, массив с четырьмя элементами целого типа можно определить как `int array[4];`);

■ инициализированный. Определяется следующим образом: `int array[]={2, 3, 5, 7};`

Ввод элементов массива

Ввод элементов одномерного массива с клавиатуры осуществляется следующим образом:

```
const int n=5;    // количество элементов в массиве
float array[n];   // объявление массива
int i;            // объявление индекса элементов
for (i=0; i<n; i++)
{ printf("array[%d]=", i); // печать подсказки
  scanf("%f", &array[i]); // считывание с клавиатуры элемента
}
```

Заполнение массива случайными числами

Пусть требуется заполнить массив равномерно распределенными случайными числами в интервале **[a,b]**. Поскольку для целых и вещественных чисел способы вычисления случайного числа в заданном интервале отличаются, рассмотрим оба варианта. Здесь и далее предполагается, что в начале программы есть строчка

`const int n = 10;`

Описание функции-датчика случайных чисел находится в заголовочном файле **stdlib.h**.

Удобно также добавить в свою программу функцию **random**:

`int random (int N) { return rand() % N; }`

которая выдает случайные числа с равномерным распределением в интервале **[0,N-1]**. Для получения случайных чисел с равномерным распределением в интервале **[a,b]** надо использовать формулу

`k = random (b - a + 1) + a;`

Для вещественных чисел формула несколько другая:

`x = rand()*(b - a)/RAND_MAX + a;`

Здесь константа **RAND_MAX** – это максимальное случайное число, которое выдает стандартная функция **rand**.

В приведенном ниже примере массив **A** заполняется случайными *целыми* числами в интервале **[-5,10]**, а массив **X** – случайными *вещественными* числами в том же интервале.

```
const int n=10;
float x[n];
int i, A[n], a=-5, b=10;;
randomize(); // инициализация генератора случайных чисел
for (i=0; i<n; i++)
    A[i]=random(b-a+1)+a; // каждому элементу массива
                        // присваивает некоторое случайное значение
for (i=0; i<n; i++)
    x[i]=(float)rand()*(b-a)/RAND_MAX+a;
```

Если необходимо, чтобы числа были не только положительными, но и отрицательными, можно записать, например, так:
array[i]=20 – random (100);

Вывод элементов массива

1. Вывод элементов массива в одну строку экрана:

```
const int n=5;
float array[n];
int i;
for (i=0; i<n; i++)
    printf(“%f \t”, array[i]); // \t – табуляция
```

Для вывода элементов массива в один столбец экрана в функции printf нужно заменить \t на \n – переход на новую строку экрана.

Операция sizeof

С помощью операции sizeof() можно определить размер памяти, которая соответствует идентификатору или типу. Операция sizeof() имеет следующий формат:

sizeof (выражение);

В качестве выражения может быть использован любой идентификатор либо имя типа. Если в качестве выражения указано имя массива, то результатом является размер всего массива (т. е. произведение числа элементов на длину типа).

```
printf(“%d \n”, sizeof (array));
```

Суммирование элементов массива

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

Удаление элемента из массива

Требуется удалить k-й элемент из массива A размером N. Удалить элемент, расположенный на k-м месте в массиве, можно, сдвинув весь «хвост» массива, начиная с (k+1)-го элемента, на одну позицию влево, т. е. выполняя операции $A_i = A_{i+1}$, $i = k, k+1, \dots, N-1$.

```
for (i=k; i<n; i++)  
    A[i]=A[i+1];
```

Включение элемента в заданную позицию массива

Перед включением элемента в k-ю позицию необходимо раздвинуть массив, т. е. передвинуть «хвост» массива вправо на одну позицию, выполняя операцию $A_{i+1} = A_i$, $i = N-1, N-2, \dots, k$. Перемещение элементов массива нужно начинать с конца. В противном случае весь «хвост» будет заполнен элементом A[k]. Далее, k-му элементу присваивается заданное значение B. Размер массива должен быть задан на 1 больше.

```
const N=6;  
void main()  
{ int A[N]={1, 2, 3, 4, 5};  
  int i, k=3, B=10;  
  clrscr();  
  for (i=N-2; i>=k; i--)  
      A[i+1]=A[i];  
  A[k]=B;
```

```

for (i=0; i<N; i++)
    printf(“%d\t”, A[i]);
getch(); }

```

В результате выполнения получится:

```

1  2  3  10  4  5

```

Перестановка элементов в массиве

Перестановка *i*-го и *j*-го элементов осуществляется с использованием вспомогательной переменной *P* (того же типа, что и элементы массива), в которую временно помещается один из элементов массива:

```

P=A[i]; A[i]=A[j]; A[j]=P;

```

Поиск минимального (максимального) элемента в массиве

Условно первый (с номером 0) элемент массива считается минимальным (максимальным), затем в цикле путем сравнения находим фактическое значение.

```

min=A[0]; // начальное значение минимума
imin=0;   // начальное значение индекса
for (i=0; i<n; i++)
    if (A[i]<min) { min=A[i]; imin=i; }
printf (“min=A[%d]=%d\n”, imin, A[imin]);

```

Если в массиве несколько элементов имеют минимальное значение, то в *imin* будет запоминаться индекс первого из них. Если проверять условие *A[i]<=min*, то – последнего.

Для поиска максимального элемента нужно в условии заменить знак *<* на знак *>*.

Примеры решения

1. Ввести массив из *n* целых чисел. Вычислить сумму (произведение) и количество элементов, кратных 3 и не кратных 2, или вывести сообщение «Таких значений нет».

```

#include <stdio.h>
#include<conio.h>

```

```

void main()
{ const int n=10;
  int a[n], s=0, k=0, i;
  // s=0; k=0; - начальное значение суммы и количества
  /* для произведения вместо s задать
  long p=1; начальное значение произведения; long, т.к. произве-
  дение быстро растёт и выходит за пределы типа int (32767) */
  clrscr();
  puts("Vvedite massiv");
  for (i=0; i<n; i++)
  { printf("a[%d]=",i+1);
    scanf("%d",&a[i]);
  }
  for (i=0; i<n; i++)
  // проверка на кратность 3 и не кратность 2 //
  if (a[i]%3==0 && a[i]%2!=0) // % - остаток от деления
  { s=s+a[i]; // накапливание суммы
    // p=p*a[i]; накапливание произведения
    k++; // накапливание количества
  }
  puts("\nResult");
  if (k==0) puts("net");
  else
  // \t – табуляция (запрограммированное число пробелов)
  printf("s=%d \t k=%d\n",s,k);
  // или для произведения
  // printf("p=%ld \t k=%d\n",p,k);
  // %ld – формат вывода данных типа long (длинный целый)
  getch();
}

```

Если функция `getch()`; не используется, то просмотреть результат можно нажатием комбинации клавиш `Alt-F5`.

Проверка, что элемент массива заканчивается заданной цифрой `m` (ввести с клавиатуры):

```

puts("Vvedite zadannoe chislo m");
scanf("%d",&m);
for (i=0; i<n; i++)

```

if (a[i] % 10 == m) /* остаток от деления на 10 равен последней цифре числа. Если этот остаток равен m, то ... */

Проверка, что элементы массива являются квадратами некоторого заданного числа **P** (здесь *обязательно требуется подключение файла math.h в разделе #include*):

```
puts("Vvedite zadannoe chislo P");
scanf("%d",&P);
for (i=0; i<n; i++)
    if (sqrt(a[i]) == P) // квадратный корень из элемента массива = P
```

2. В одномерном массиве из n элементов найти и вывести значение и номер первого отрицательного значения или вывести сообщение об отсутствии отрицательных чисел.

При решении данной задачи может потребоваться досрочный выход из цикла (при наличии отрицательного числа), т. е. оператор break.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void main()
{ const int n=10;
  int a[n], i;
  int f=0; // флажок указывает на успешность поиска
           // изначально 0, т. к. еще ничего не найдено
  // заполним массив случайными числами
  randomize();
  for (i=0; i<n; i++)
  { a[i]=20 - random(100);
    printf ("%d\t", a[i]);
  }
  printf ("\n"); // перевод курсора на новую строку экрана
  // начало поиска
  for (i=0; i<n; i++)
  if (a[i]<0)
  { f=1; //устанавливаем флажок в состояние «истина»
    printf ("a[%d]=%d\n", i, a[i]);
```

```

        break;
    }
    if (f==0) puts (“<0 нет”);
    getch();
}

```

Для поиска значения и номера последнего отрицательного значения массив лучше рассматривать с конца, т. е. при поиске записать `for (i=n-1; i>=0; i--)`

Порядок выполнения работы

1. Изучить теоретические сведения.
2. Ответить на контрольные вопросы.
3. Выполнить задание.

Контрольные вопросы

1. Что такое массив?
2. Почему индекс последнего элемента массива на 1 меньше его размера?
3. Что в записи `A[4]=12` является именем массива, что – индексом, а что – значением элемента?
4. Какой специальный символ используется для вывода данных в строку экрана?
5. Почему при поиске какого-либо элемента в массиве нельзя обойтись без цикла?

Задания для выполнения

Часть 1.

1. Ввести одномерный массив из **n** элементов. Вычислить среднее арифметическое значение элементов массива, имеющих четные отрицательные значения, или вывести сообщение об отсутствии в массиве таких значений.

2. Ввести одномерный массив из **n** элементов. Вычислить произведение и количество элементов, находящихся в интервале от **A** до **B**, или вывести сообщение об отсутствии в массиве таких значений.

3. Ввести массив из **n** целых чисел. Вычислить среднее арифметическое значение элементов, являющихся нечётными числами, или вывести сообщение «Таких значений нет».

4. Ввести массив из n целых чисел. Вычислить произведение и количество элементов, кратных 3 и некратных 5, или вывести сообщение «Таких значений нет».

5. Ввести массив из n целых чисел. Найти и вывести те элементы, которые при делении на 7 дают остаток 2 или 5, или вывести сообщение «Таких значений нет».

6. Ввести массив из n целых чисел. Найти количество и сумму тех элементов, которые делятся на 5 и не делятся на 7, или вывести сообщение «Таких значений нет».

7. Ввести массив из n целых чисел. Вычислить произведение элементов, кратных заданному числу m (ввести с клавиатуры), или вывести сообщение «Таких значений нет».

8. Ввести массив из n элементов. Получить сумму и количество элементов, принадлежащих отрезку $[3, 17]$, или вывести сообщение «Таких значений нет».

9. Ввести массив A из n элементов. Найти и вывести минимальное из A_0, A_2, A_4, \dots и максимальное из A_1, A_3, A_5, \dots

10. Ввести массив из n элементов. Получить сумму тех членов массива, которые нечётны и положительны, или вывести сообщение «Таких значений нет».

11. Ввести массив из n целых чисел. Вычислить среднее арифметическое значение элементов, порядковые номера которых совпадают со значением этого элемента, или вывести сообщение «Таких значений нет».

12. Ввести массив из n целых чисел. Вывести те значения, которые заканчиваются заданной цифрой (ввести с клавиатуры), или вывести сообщение «Таких значений нет».

13. Ввести массив из n целых чисел. Найти сумму тех из них, которые являются квадратами некоторого заданного числа m , или вывести сообщение «Таких значений нет».

14. Ввести массив из n элементов и некоторое число P . Вывести номера тех элементов, модуль значений которых больше этого числа, или вывести сообщение «Таких значений нет».

15. Ввести массив из n элементов, в котором есть по крайней мере два отрицательных числа. Вычислить произведение и количество элементов, находящихся между первым и последним отрицательным значениями.

Часть 2.

1. Ввести массив из **n** элементов и два индекса. Поменять местами элементы, находящиеся под данными индексами. Вывести преобразованный массив.

2. Ввести массив из **n** вещественных чисел. Вычислить их среднее арифметическое значение и найти элемент, значение которого наиболее близко к среднему арифметическому.

3. Ввести массив из **n** элементов, содержащий хотя бы одно отрицательное значение. Вычислить среднее арифметическое значение чисел, расположенных после первого отрицательного элемента.

4. Ввести массив из **n** элементов. Найти в нем максимальное значение. Заменить значения элементов массива путем вычитания их из максимального. Вывести преобразованный массив.

5. Ввести массив **n** вещественных чисел. Найти в нем минимальное значение и разделить на него каждый элемент. Вывести преобразованный массив или сообщение «min=0».

6. Ввести массив из **n** вещественных чисел и числа **a** и **b**. Заменить нулями все элементы, значения которых не принадлежат отрезку [**a**, **b**], или вывести сообщение «Таких значений нет».

7. Ввести массив из **n** элементов, в котором есть хотя бы одно положительное значение. Вычислить произведение и количество элементов, находящихся до первого положительного. Если положительное значение первое, то вывести сообщение «A[0]>0».

8. Ввести массив из **n** различных целых чисел. Поменять местами наибольшее и наименьшее значения. Вывести преобразованный массив.

9. Ввести массив из **n** элементов и некоторое число **P**. Заменить все, большие **P**, элементы массива этим числом. Вычислить количество таких элементов. Если оно больше нуля, вывести преобразованный массив.

10. Ввести массив из **n** элементов. Среди них есть равные. Найти его первое наибольшее значение и заменить его нулем. Вывести преобразованный массив.

11. Ввести массив из **n** элементов. Вычислить сумму модулей элементов массива, расположенных после минимального по модулю элемента.

12. Ввести массив из n вещественных чисел. Все отрицательные числа увеличить на 0.5, а все неотрицательные на 0.1. Вывести преобразованный массив.

13. Ввести массив из n вещественных чисел. Вычислить их среднее арифметическое значение и вычесть его из каждого элемента. Вывести преобразованный массив.

14. Ввести массив из n целых чисел. Заменить нулями элементы, модуль которых при делении на p даёт в остатке q . (Значения p и q ввести с клавиатуры). Вывести преобразованный массив или сообщение «Таких значений нет».

15. Ввести массив из n различных чисел. Найти в нем максимальное значение и поменять его местами со вторым элементом. Вывести преобразованный массив.