

Лабораторная работа №6

«Массивы»

Теоретическое введение

Массив — это *несколько смежных* ячеек оперативной памяти, имеющих *один и тот же тип* и объединенных одним, общим именем.

Объявление массива целых чисел на 20 ячеек:

```
int my_array[20];
```

Инициализация всех элементов массива нулем:

```
for(int i=0; i<20; i++){  
    my_array[i] = 0;           // i = 0..19  
}
```

Вывод массива на экран в виде таблицы: «№ ячейки — значение»:

```
cout << "№яч.\tзнач.\n";  
for(int i=0; i<20; i++){  
    cout << i << "\t" << my_array[i] << "\n";  
}
```

Основное отличие массива на 20 ячеек от набора из 20 отдельных переменных `c0`, `c1`, `c2...c19` в том, что при обращении к ячейкам массива мы можем не только указать конкретный номер ячейки (например, 8):

```
cout << my_array[8];
```

но и **вычислить** этот номер, исходя из каких-то других переменных в программе:

```
int i;  
cout << "Какая ячейка интересует: ";  
cin >> i;  
cout << "Соседи Вашей ячейки имеют следующие значения:\n";  
cout << "Левый сосед: " << my_array[i-1] << "\n";  
cout << "Правый сосед: " << my_array[i+1] << "\n";
```

По этой причине вывод на экран всех элементов массива может быть осуществлен с помощью цикла (см. выше), а вывод на экран 20 отдельных переменных `c0`, `c1`, `c2...c19` потребует написания *20 отдельных операторов вывода*:

```
cout << c0 << "\n";  
cout << c1 << "\n";  
...  
cout << c19 << "\n";
```

Основные правила работы с массивами

1. Размер массива, указываемый при его объявлении, может быть *только константой*. Это значит, что нельзя спросить у пользователя, сколько элементов должно быть в массиве, и потом создать массив под введенное с клавиатуры число элементов¹.
2. Ячейки массива имеют номера от 0 до $N-1$, где N — общее количество элементов.
3. Локальный массив, объявленный внутри функции (например, внутри функции `main()`) по умолчанию никак не инициализирован и содержит «мусор». В большинстве случаев такие массивы надо явно заполнять нулями с помощью цикла. Глобальные массивы автоматически заполняются нулями.

¹ Решить эту проблему помогает динамическая память.

Задания для самостоятельного решения

Задание 1. Объявить массив на 20 элементов типа `int`, заполнить его квадратами целых чисел от 1 до 20 и вывести на экран в виде следующей таблицы:

n	n*n
1	1
2	4
3	9
<i>и так далее</i>	
20	400

(см. ниже раздел «Примеры программ»)

Задание 2. Вы — начинающий кинорежиссер и только что завершили работу над своим дебютным проектом. Вскоре Вам предстоит премьерный показ, на котором Вы планируете попросить всех гостей (а всего их 20 человек) выставить Вашему фильму оценку по стандартной imdb-шной шкале — от 1 до 10 баллов. Для этого Вам нужна компьютерная программа, которая:

1. Прочитает с клавиатуры 20 оценок, выставленных гостями.
2. Выведет на экран таблицу вида: «оценка — количество гостей, выставивших эту оценку».

Например, эта таблица могла бы выглядеть следующим образом:

оценка	количество голосов
1	3
2	0
3	0
4	0
5	1
7	3
8	5
9	5
10	3

Очевидно, сумма элементов правого столбца таблицы должна равняться 20. Для хранения такой таблицы удобен массив на 10 целых чисел, в котором в n -ой ячейке хранится количество гостей, выставивших оценку $n+1$ (т.к. номера ячеек начинаются с 0, а оценки — с 1). Остальные подробности по данной программе см. в **4-ой главе книги Дейтела**. Задание — написать такую программу.

Задание 3. Выполните свой вариант индивидуального задания №6 (см. Приложение).

Задание 4. По окончании премьерного показа Вы, как режиссер, решили собрать некоторых из своих гостей в специальной темной комнате для дополнительной беседы. В число «избранных» войдут те, кто поставил Вашему фильму низкую оценку...те, кто не очень лестно отзывался о нем в кулуарах...в общем Вы еще не составили конкретный список и поэтому, возможно, еще будете его редактировать. Однако Вы уже решили, что в этом списке — из соображений безопасности — не будет конкретных фамилий. Вместо этого список будет содержать номера пригласительных билетов Ваших гостей (они пронумерованы от 1 до 20). В связи с этим, Вам нужна *программа*, которая:

- хранит в массиве номера всех приглашенных для доп. беседы;
- позволяет просмотреть текущее содержимое массива;
- позволяет добавить в заданную позицию массива нового гостя (с заданным номером);
- позволяет удалить гостя из заданной позиции в массиве.

Технические подробности: так как всего гостей 20, то массива на 20 элементов нам вполне хватит для хранения списка «избранных» любой длины.

См. также раздел «Примеры программ».

Пример выполнения:

Выберите:

1. Показать список.
2. Добавить.
3. Удалить.
4. Выход.

1

Список:

1. 15
2. 11
3. 2
4. 6

Выберите:

1. Показать список.
2. Добавить.
3. Удалить.
4. Выход.

2

Введите элемент: 19

Введите позицию: 2

Добавлено!

Выберите:

1. Показать список.
2. Добавить.
3. Удалить.
4. Выход.

1

Список:

1. 15
2. 19
3. 11
4. 2
5. 6

Задание 5. Выполните свой вариант индивидуального задания №7 (см. Приложение).

Примеры программ

Поиск в массиве максимального элемента

```
#include <iostream>
using namespace std;

const int N = 20;
int arr[N];

int main(){

    // 1 fill the array
    // with randoms in range 1..100
    for(int i=0; i<N; i++){
        arr[i] = rand() % 100 + 1;
    } // for

    // 2 find max
    int max_value = arr[0];
    for(int i=1; i<N; i++){
        if(arr[i] > max_value)
            max_value = arr[i];
    }

    // 3 print max
    cout << "Max element = " << max_value << "\n";

    return 0;
}
```

Удаление элемента из массива

```
#include <iostream>
using namespace std;

const int N = 20;           // total amount
int arr[N];
int N_used = 0;             // number of used

int main()
{
    // fill some
    N_used = 4;
    for(int i=0; i<N_used; i++){
        arr[i] = rand()%20 + 1;
    }

    for(;;){

        // show
        for(int i=0; i<N_used; i++){
            cout << i+1 << ". " << arr[i] << "\n";
        }

        // ask element to delete
        cout << "Which one to delete? ";
        int pos;
        cin>>pos;
        pos--;               // we begin from 0

        // check bounds
        if(pos<0 || pos>=N_used){
```

```
        cout << "Wrong choice!\n";
        continue;
    }

    // delete!
    for(int i=pos; i<N_used-1; i++)
        arr[i] = arr[i+1];
    N_used--;
}

return 0;
}
```

Контрольные вопросы

1. Какие номера имеют элементы массива, если массив — на N элементов?
2. Сколько памяти занимает массив на 1000 элементов типа int?
3. Какие данные — на входе, а какие — на выходе каждой из программ данной лабораторной?
4. Как изменить размер массива?
5. Как сгенерировать случайное число в диапазоне от 500 до 512?

Оценивание

Содержание отчета

1. Условия заданий.
2. Исходные коды.
3. Скриншоты выполнения.

Баллы за задания

Задание	Баллы	
1	обязательное	1
2	обязательное	1
3		2
4		2
5		2
Всего		8

Бонусы

Досрочная сдача: +1 балл.

Несвоевременная сдача: макс. балл уменьшается на 2 за каждую просроченную неделю.

На следующие бонусные задания ограничение по времени — 2 недели!

«Быки и коровы»

Компьютер *загадывает* четырехзначное число с неповторяющимися цифрами, а пользователь пытается его отгадать, следуя подсказкам компьютера. За хорошую реализацию — 3 балла. За хорошую реализацию алгоритма *отгадывания* числа — по договоренности.

http://ru.wikipedia.org/wiki/Быки_и_коровы

«Длинная арифметика»

Вместо лабораторной напишите программу, которая позволяет складывать, вычитать, умножать и делить целые числа произвольной длины (примерно так, как нас учили это делать в начальной школе). За хорошую реализацию — 5 баллов. За хорошую реализацию того же, но для десятичных дробей — по договоренности. Также интересует возможность вычисления с произвольной точностью тригонометрических, показательной и логарифмической функции (см. ряды Тейлора и цепные дроби).

Приложение: индивидуальные задания

Индивидуальное задание №6

Примечания:

1. Все упомянутые в заданиях массивы вводятся с клавиатуры.
2. Размер массива N задается в виде константы в исходном коде программы.
3. Когда речь идет о «четных» или «нечетных» номерах, имеется в виду, что нумерация начинается с единицы.

Задания:

1. Дан целочисленный массив размера N . Вывести все содержащиеся в данном массиве четные числа в порядке убывания их номеров, а также их количество.
2. Дан целочисленный массив A размера N . Вывести вначале его элементы с нечетными номерами в порядке возрастания номеров, а затем — элементы с четными номерами в порядке убывания номеров:
 $A_1, A_3, A_5, \dots, A_6, A_4, A_2$.
3. Дан целочисленный массив A размера N . Найти минимальный элемент из его элементов с четными номерами: A_2, A_4, A_6, \dots .
4. Дан целочисленный массив A размера N . Найти номера тех элементов массива, которые больше своего правого соседа, и количество таких элементов. Найденные номера выводить в порядке их возрастания.
5. Дан целочисленный массив размера N . Найти сумму тех элементов, которые расположены между максимальным и минимальным элементами (в сумму включить и оба этих элемента).
6. Дан целочисленный массив размера N . Найти длину самой длинной последовательности подряд идущих элементов, равных 0.
7. Дан целочисленный массив размера N . Найти сумму неотрицательных элементов массива.
8. Дан целочисленный массив размера N . Найти сумму элементов массива с четными номерами.
9. Дан целочисленный массив размера N . Найти сумму элементов массива, расположенных между первым и последним отрицательными элементами.
10. Дан целочисленный массив размера N . Найти сумму элементов массива, расположенных до последнего положительного элемента.
11. Дан целочисленный массив размера N . Найти максимальный по модулю элемент массива.
12. Дан целочисленный массив размера N . Найти произведение элементов массива, расположенных между первым и вторым нулевыми элементами.
13. Дан целочисленный массив размера N . Найти сумму элементов массива, расположенных между i -м и j -м положительными элементами.
14. Дан целочисленный массив размера N . Найти сумму модулей элементов массива, расположенных после первого элемента, равного нулю.
15. Дан целочисленный массив размера N . Найти номер максимального по модулю элемента массива.
16. Дан целочисленный массив размера N . Найти сумму элементов массива, расположенных после первого положительного элемента.
17. Дан целочисленный массив размера N . Найти сумму элементов массива, расположенных после минимального элемента.
18. Дан целочисленный массив размера N . Найти сумму модулей элементов массива, расположенных после минимального по модулю элемента.
19. Дан целочисленный массив размера N . Найти сумму элементов массива, расположенных после последнего элемента, равного нулю.
20. Дан целочисленный массив размера N . Найти сумму положительных элементов массива, расположенных до максимального элемента.

Индивидуальное задание №7

Примечания:

1. Все упомянутые в заданиях массивы вводятся с клавиатуры.
2. После преобразования массив-результат выводится на экран.
3. Размер массива N задается в виде константы в исходном коде программы.
4. Когда речь идет о «четных», «нечетных» или номерах K и L , то имеется в виду, что нумерация начинается с единицы.
5. Когда из массива «удаляются элементы», то его размер, очевидно, *меньше не становится*. Просто появляются неиспользуемые ячейки. В таких случаях на экран надо выводить только ту часть массива, в которой остались «неудаленные» элементы.
6. В заданиях, в которых требуется добавить элементы в массив, массив нужно заранее создать на несколько большее количество элементов — «с запасом».

Задания:

1. Дан целочисленный массив размера N . Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все остальные.
2. Дан целочисленный массив размера N . Преобразовать массив таким образом, чтобы сначала располагались все элементы равные нулю, а потом — все остальные.
3. Дан целочисленный массив размера N . Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине — элементы, стоявшие в четных позициях.
4. Дан целочисленный массив размера N . Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом — все остальные.
5. Дан целочисленный массив размера N и целые числа K и L ($1 \leq K < L \leq N$). Удалить из массива элементы с номерами от K до L включительно.
6. Дан целочисленный массив размера N . Удалить из массива все элементы с четными номерами.
7. Дан целочисленный массив размера N . Сжать массив, удалив из него все элементы, модуль которых не превышает 1.
8. Дан целочисленный массив размера N и два целых числа K и M ($1 \leq K \leq N$, $1 \leq M \leq 10$). Перед элементом массива с номером K вставить M новых элементов с нулевыми значениями.
9. Дан целочисленный массив размера N . Утроить в нем вхождения всех элементов с нечетными номерами.
10. Дан целочисленный массив размера N . Удалить из массива все одинаковые элементы, оставив их первые вхождения.
11. Дан целочисленный массив размера N . Удалить из массива все элементы, встречающиеся более трех раз.
12. Дан целочисленный массив размера N . Удалить из массива все элементы, встречающиеся ровно два раза.