

ЗАДАНИЕ №1

ОДНОМЕРНЫЕ МАССИВЫ И УКАЗАТЕЛИ

ЦЕЛЬ РАБОТЫ: изучить принципы написания программ, использующих одномерные массивы и указатели на языке программирования C++.

ПРИМЕРЫ ПРОГРАММ

Пример 2.1. Количество элементов между минимумом и максимумом

Написать программу, которая для целочисленного массива из 100 элементов определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

```
#include <iostream>
#include <locale>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    const int n = 10; int a[n];
    cout << "Введите " << n << " целых чисел:" << endl;
    for (int i = 0; i < n; i++) cin >> a[i];
    for (int i = 0; i < n; i++) cout << a[i] << " ";
    int imax = 0, imin = 0, count = 0; for (int i = imax = imin = 0; i < n; i++) {
        if (a[i] > a[imax]) imax = i;
        if (a[i] < a[imin]) imin = i;
    }
    int d = imax < imin ? 1 : imax > imin ? -1 : 0;
    for (int i = imax + d; i != imin; i += d)
        if (a[i] > 0) count++;
    cout << "Количество положительных: " << count << endl;
    system("pause");
}
```

Пример 2.2. Сумма элементов правее последнего отрицательного

Написать программу, которая для вещественного массива из n элементов определяет сумму его элементов, расположенных правее последнего отрицательного элемента.

```
#include <iostream>
#include <locale>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    int n;
```

```

cout << "Введите количество элементов: "; cin >> n;
double *a = new double[n];
cout << "Введите элементы массива: ";
for (int i = 0; i < n; i++) cin >> a[i];
int ineg = -1;
for (int i = 0; i < n; i++) if (a[i] < 0) ineg = i;
if (ineg == -1)    cout << "Отрицательных элементов нет\n";
else if (ineg == n - 1) cout << "Отрицательный элемент - последний\n";
else {
    double sum = 0;
    for (int i = ineg + 1; i < n; i++) sum += a[i];
    cout << "Сумма отрицательных: \n" << sum;
}
system("pause");
}

```

Пример 2.3. Сортировка массива методом выбора

Написать программу, которая упорядочивает вещественный массив по возрастанию значений его элементов методом выбора.

```

#include <iostream>
#include <cmath>
#include <locale>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    int n;
    cout << "Введите количество элементов: "; cin >> n;
    double *arr = new double[n];
    cout << "Введите элементы массива: ";
    for (int i = 0; i < n; i++) cin >> arr[i];
    /////////////////////////////////// Сортировка методом выбора ///////////////////////////////////
    double tmp;
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[j] < arr[i]) {
                tmp = arr[i];
                arr[i] = arr[j];
                arr[j] = tmp;
            }
        }
    }
    cout << "Отсортированный по возрастанию массив: "<<endl;
    for (int i = 0; i < n; i++) cout << arr[i] << ' ';cout << endl;
    system("pause");
}

```

```
}
```

Пример 2.4. Быстрая сортировка массива

Написать программу, которая упорядочивает вещественный массив по возрастанию значений его элементов методом быстрой сортировки.

```
#include <iostream>
#include <cmath>
#include <locale>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    const int n = 20; float arr[n];
    int *stackl = new int[n], *stackr = new int[n];
    cout << "Введите элементы массива: ";
    for (int i = 0; i < n; i++) cin >> arr[i];
    //////////////// Быстрая сортировка //////////////////////
    int sp = 1; stackl[sp] = 0; stackr[sp] = n - 1; // 1
    while (sp > 0) { // Выборка из стека последнего запроса // 2
        int left = stackl[sp]; // 3
        int right = stackr[sp]; // 4
        sp--; // 5
        while (left < right) { // 6
            // Разделение { arr[left] .. arr[right] }
            int i = left, j = right; // 7
            float middle = arr[(left + right) / 2]; // 8
            while (i < j) { // 9
                while (arr[i] < middle) i++; // 10
                while (middle < arr[j]) j--; // 11
                if (i <= j) {
                    float temp = arr[i]; arr[i] = arr[j]; arr[j] = temp; i++; j--;
                }
            }
            if (i < right) { // Запись в стек запроса из правой части // 12
                sp++;
                stackl[sp] = i; stackr[sp] = right;
            }
            right = j; // Теперь left и right ограничивают левую часть // 13
        }
    }
    for (int i = 0; i < n; i++) cout << arr[i] << ' ';
    system("pause");
}
```

На каждом шаге сортируется один фрагмент массива. Левая и правая границы фрагмента хранятся в переменных *left* и *right*. Сначала фрагмент устанавливается размером с весь массив (*строка 1*). В *операторе 8* выбирается «средний» элемент. Для продвижения по массиву слева направо используется переменная *i* (*цикл 10*), справа налево – переменная *j* (*цикл 11*). Их начальные значения устанавливаются в *операторе 7*. После того, как оба счетчика «сойдутся» где-то в средней части массива, происходит выход из *цикла 9* на *оператор 12*, в котором в стек заносятся границы правой части фрагмента. В *операторе 13* устанавливаются новые границы левой части для сортировки на следующем шаге. Если фрагмент уже настолько мал, что сортировать его не требуется, происходит выход из *цикла 6*, после чего выполняется выборка из стека границ еще не отсортированного фрагмента (*операторы 3, 4*). Если стек пуст, происходит выход из главного *цикла 2*. Массив отсортирован.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. Разработать на языке C++ в соответствии со своим вариантом программу, использующую одномерные массивы и указатели. В качестве исходных данных используется массив, состоящий из *n* вещественных величин. При выполнении работы рекомендуется использовать динамические массивы, хотя допустимы и статические, размерность которых задается константой. Для сортировки массива желательно использовать метод быстрой сортировки, допустимо использовать более простые методы, например метод выбора.
2. Номер варианта определяется по последними двум цифрам зачетной книжки согласно табл.2.1.

Таблица 2.1 – Формулы расчета варианта исходных данных

Последние две цифры зачетной книжки, Nзач	Номер варианта, Nвар
Nзач=от 1 до 20	Nвар=Nзач
Nзач=от 21 до 40	Nвар=Nзач-20
Nзач=от 41 до 60	Nвар=Nзач-40
Nзач=от 61 до 80	Nвар=Nзач-60
Nзач=от 81 до 99	Nвар=Nзач-80

3. В коде должны присутствовать комментарии, поясняющие алгоритм работы программы.
4. Выполненное задание предоставить в виде файла с расширением .cpp. В начале файла написать в виде комментариев: номер и вид работы, тему работы, Ф.И.О., группу, вариант задания, задание.
5. Подготовиться к защите и защитить работу у преподавателя.

ВАРИАНТЫ ИСХОДНЫХ ДАННЫХ

Вариант 1

- ☐ Найти сумму отрицательных элементов массива.
- ☐ Найти произведение элементов массива, расположенных между максимальным и минимальным элементами.
- ☐ Упорядочить элементы массива по возрастанию.

Вариант 2

- ☐ Найти сумму положительных элементов массива.
- ☐ Найти произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.
- ☐ Упорядочить элементы массива по убыванию.

Вариант 3

- ☐ Найти произведение элементов массива с четными номерами.
- ☐ Найти сумму элементов массива, расположенных между первым и последним нулевыми элементами.
- ☐ Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все отрицательные (элементы, равные 0, считать положительными).

Вариант 4

- ☐ Найти сумму элементов массива с нечетными номерами.
- ☐ Найти сумму элементов массива, расположенных между первым и последним отрицательными элементами.
- ☐ Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

Вариант 5

- ☐ Найти максимальный элемент массива.
- ☐ Найти сумму элементов массива, расположенных до последнего положительного элемента.
- ☐ Сжать массив, удалив из него все элементы, модуль которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.

Вариант 6

- ☐ Найти минимальный элемент массива.
- ☐ Найти сумму элементов массива, расположенных между первым и последним положительными элементами.

- ☐ Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом — все остальные.

Вариант 7

- ☐ Найти номер максимального элемента массива.
- ☐ Найти произведение элементов массива, расположенных между первым и вторым нулевыми элементами.
- ☐ Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине — элементы, стоявшие в четных позициях.

Вариант 8

- ☐ Найти номер минимального элемента массива.
- ☐ Найти сумму элементов массива, расположенных между первым и вторым отрицательными элементами.
- ☐ Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом — все остальные.

Вариант 9

- ☐ Найти максимальный по модулю элемент массива.
- ☐ Найти сумму элементов массива, расположенных между первым и вторым положительными элементами.
- ☐ Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

Вариант 10

- ☐ Найти минимальный по модулю элемент массива.
- ☐ Найти сумму модулей элементов массива, расположенных после первого элемента, равного нулю.
- ☐ Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине — элементы, стоявшие в нечетных позициях.

Вариант 11

- ☐ Найти номер минимального по модулю элемента массива.
- ☐ Найти сумму модулей элементов массива, расположенных после первого отрицательного элемента.
- ☐ Сжать массив, удалив из него все элементы, величина которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.

Вариант 12

- ☐ Найти номер максимального по модулю элемента массива.
- ☐ Найти сумму элементов массива, расположенных после первого положительного элемента.
- ☐ Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале $[a, b]$, а потом — все остальные.

Вариант 13

- ☐ Найти количество элементов массива, лежащих в диапазоне от A до B .
- ☐ Найти сумму элементов массива, расположенных после максимального элемента.
- ☐ Упорядочить элементы массива по убыванию модулей элементов.

Вариант 14

- ☐ Найти количество элементов массива, равных 0.
- ☐ Найти сумму элементов массива, расположенных после минимального элемента.
- ☐ Упорядочить элементы массива по возрастанию модулей элементов.

Вариант 15

- ☐ Найти количество элементов массива, больших C .
- ☐ Найти произведение элементов массива, расположенных после максимального по модулю элемента.
- ☐ Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом — все положительные (элементы, равные 0, считать положительными).

Вариант 16

- ☐ Найти количество отрицательных элементов массива.
- ☐ Найти сумму модулей элементов массива, расположенных после минимального по модулю элемента.
- ☐ Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

Вариант 17

- ☐ Найти количество положительных элементов массива.
- ☐ Найти сумму элементов массива, расположенных после последнего элемента, равного нулю.
- ☐ Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом — все остальные.

Вариант 18

- ☐ Найти количество элементов массива, меньших C .
- ☐ Найти сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.
- ☐ Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20%, а потом — все остальные.

Вариант 19

- ☐ Найти произведение отрицательных элементов массива.
- ☐ Найти сумму положительных элементов массива, расположенных до максимального элемента.
- ☐ Изменить порядок следования элементов в массиве на обратный.

Вариант 20

- ☐ Найти произведение положительных элементов массива.
- ☐ Найти сумму элементов массива, расположенных до минимального элемента.
- ☐ Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего используются указатели?
2. Как производится объявление указателей?
3. Какие операции можно производить над указателями?
4. В чем отличия указателей и ссылок?
5. Что такое массив?
6. В чем сходства и различия статических и динамических массивов?
7. Какие методы сортировки массивов вы знаете?
8. В чем сущность метода сортировки выбором?
9. В чем сущность метода сортировки обменом?
10. В чем сущность метода сортировки вставками?
11. В чем сущность метода быстрой сортировки?