

## Лабораторная работа №9

**Тема:** Разработка и отладка алгоритмов и программ с использованием указателей.

**Цель занятия:** Выработать умения и навыки работы со средой программирования Microsoft Visual Studio.

**Время выполнения:** 2 часа.

### *Содержание работы и последовательность ее исполнения.*

1. Запустить Microsoft Visual Studio выбрав пункт меню «Пуск → Все программы → Microsoft Visual Studio → Microsoft Visual Studio».
2. Для каждого задания создавать **отдельный пустой консольный проект**.

Дано три массива размером из 20 элементов каждый. Значениями каждого из массивов являются случайные целые числа в диапазоне от -10 до 10.

#### **Задание 1**

Написать функцию, которая принимает в качестве параметров два указателя на массивы и возвращает указатель на массив, удовлетворяющий условию:

- Вариант 1:** Сумма элементов массива должна быть максимальной.
  - Вариант 2:** Количество положительных элементов массива должно быть максимальным.
  - Вариант 3:** Среднее арифметическое элементов массива должно быть минимальным.
  - Вариант 4:** Количество отрицательных элементов должно быть меньше положительных.
  - Вариант 5:** Количество элементов меньших справа стоящих должно быть минимальным.
  - Вариант 6:** Количество отрицательных элементов массива должно быть минимальным.
  - Вариант 7:** Среднее арифметическое массива должно быть наиболее приближенным к нулю.
  - Вариант 8:** Сумма элементов массива должна быть минимальной.
  - Вариант 9:** Количество положительных элементов массива должно быть минимальным.
  - Вариант 10:** Количество отрицательных элементов должно быть больше положительных.
  - Вариант 11:** Среднее арифметическое элементов массива должно быть максимальным.
  - Вариант 12:** Разница сумм элементов на четных позициях и элементов на нечетных позициях должна быть минимальной.
  - Вариант 13:** Количество отрицательных элементов массива должно быть максимальным.
  - Вариант 14:** Количество элементов больших слева стоящих должно быть максимальным.
  - Вариант 15:** Среднее арифметическое массива должно быть наиболее отдаленным от нуля.
- С помощью написанной процедуры выбрать получить указатель одного из трех массивов.

#### **Задание 2**

Выбранный массив отсортировать по возрастанию (для четных вариантов) или по убыванию (для нечетных вариантов) и вывести на экран.

#### **Примечание к выполнению обоих заданий:**

- 1) все промежуточные вычисления должны быть оформлены в виде отдельных функций.
- 2) все массивы, как начальные так и найденный, должны быть выведены на экран.

#### **Пример выполнения:**

Дано три массива размером из 20 элементов каждый. Значениями каждого из массивов являются случайные целые числа в диапазоне от -10 до 10.

Написать функцию, которая принимает в качестве параметров два указателя на массивы и возвращает указатель на массив, удовлетворяющий условию: количество элементов массива меньших своих соседей должно быть минимальным.

Найденный массив сортировать по следующему принципу: первые десять элементов по возрастанию, последующие десять элементов по убыванию.

```

#include <stdio.h> // стандартный ввод/вывод
#include <time.h> // функции работы с таймером
#include <stdlib.h> // библиотека стандартных функций
// функция заполнения массива случайными числами, принимает указатель на массив
// для заполнения
void fillArray(int *mas) {
    for(int i=0; i<20; i++) // цикл по всем элементам массива
        mas[i]=rand()%21-10; // генерируем случайное число
}
// функция выводит на экран содержимое массива
void printArray(int *mas) {
    printf("{");
    for(int i=0; i<20; i++)
        printf("%3i",mas[i]);
    printf("}\n");
}
// функция вычисляем количество элементов массива меньших своих соседей
int calcElems(int *mas) {
    int cnt=0; // временная переменная-счетчик
    for(int i=1; i<19; i++) // цикл по всем элементам кроме крайних
        if(mas[i]<mas[i-1] && mas[i]>mas[i+1]) // проверяем условие
            cnt++; // инкрементируем счетчик если условие выполняется
    return cnt; // выходим из функции и возвращаем значение
}
// функция возвращаем указатель на массив, в котором количество элементов меньших
// своих соседей минимально
int *searchArray(int *mas1,int *mas2) {
    int cnt1=calcElems(mas1), // считаем количество элементов в первом массиве
        cnt2=calcElems(mas2); // считаем количество элементов во втором массиве
    if(cnt1<cnt2) // сравниваем результаты
        return mas1; // возвращаем указатель на массив...
    else // ...в зависимости от...
        return mas2; // ...условия
}
// функция обмениваем местами значения ячеек памяти по указателям а и b
void swap(int *a,int *b) {
    int tmp=*a;
    *a=*b;
    *b=tmp;
}
// функция сортировки массива методом пузырька
void sortArray(int *mas)
{
    for(int i=0; i<10; i++)
        for(int j=0; j<10; j++) {
            if(mas[i] < mas[j]) swap(&mas[i],&mas[j]);
            if(mas[i+10]>mas[j+10]) swap(&mas[i+10],&mas[j+10]);
        }
}
// головная функция
int main() {
    srand((unsigned)time(NULL)); // инициализация счетчика случайных чисел
    int m1[20],m2[20],m3[20]; // описание трех массивов
    // последовательно заполняем массивы случайными числами и
    // выводим их на экран
    fillArray(m1); printf("M1="); printArray(m1);
    fillArray(m2); printf("M2="); printArray(m2);
    fillArray(m3); printf("M3="); printArray(m3);
    // за два захода выбираем массив удовлетворяющий условию
    int *min=searchArray(m1,m2);
    min=searchArray(min,m3);
    printf("MN="); printArray(min); // и выводим его на экран
    sortArray(min); // сортируем массив
    printf("ST="); printArray(min); // и выводим на экран
    return 0; // выходим из программы
}

```