

Лабораторная работа №7

Тема: Разработка и отладка алгоритмов и программ с применением пользовательских функций.

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**.

Задание 1. Функции и массивы

Выполнить задания лабораторной работы №4 («Разработка и отладка программ обработки массивов»), оформив каждый пункт задания в виде функции. Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Задание 2. Функции и строки

Выполнить задание номер 3 лабораторной работы №6 («Разработка и отладка алгоритмов и программ по обработке строк символов»), оформив каждый пункт задания в виде функции. Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Пример выполнения:

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- количество отрицательных элементов массива;
- сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

// функция для заполнения массива m случайными n числами
void fillArray(int *m,int n) {
    for(int i=0; i<n; i++)
        m[i]=rand()%50-25;
}

// функция выводит содержимое массива на экран
void showArray(int *m,int n)
{
    for(int i=0; i<n; i++)
        printf("%4i",m[i]);
    printf("\n");
}

// функция подсчитываем количество отрицательных элементов массива a
int negateCount(int *a,int cnt)
{
    int c=0;
    for(int i=0; i<cnt; i++)
        if(a[i]<0) c++;
    return c;
}

// функция для поиска последнего отрицательного числа
int minimumSearch(int *b,int c) {
    int min=0;
    for(int i=1; i<c; i++)
        if(abs(b[min])>abs(b[i])) min=i;
    return min;
}
```

```

// функция находит сумму элементов массива а начиная с позиции m+1
int summArray(int *a,int b,int m) {
    int summ=0;
    for(int i=m+1; i<b; i++)
        summ+=abs(a[i]);
    return summ;
}

// функция заменяем все отрицательные элементы массива на их квадрат
void negateCorrect(int *m,int n) {
    for(int i=0; i<n; i++)
        if(m[i]<0) m[i]*=m[i];
}

// функция сортирует массив
void sortArray(int *a,int c) {
    for(int i=0; i<c; i++)
        for(int j=0; j<c; j++)
            if(a[i]<a[j]) {
                int tmp=a[i]; a[i]=a[j]; a[j]=tmp;
            }
}

int main() {
    srand((unsigned)time(NULL));

    int n;
    // вводим размер массива
    printf("N=");
    scanf("%i",&n);

    // создаем массив
    int m[100];

    // заполнение массива
    fillArray(m,n);
    printf("Начальный массив:\n");
    showArray(m,n);
    // подсчитываем кол-во отрицательных чисел
    printf("Кол-во отрицательных чисел: %i\n",negateCount(m,n));
    // поиск последнего минимального по модулю элемента
    int min=minumumSearch(m,n);
    printf("Позиция последнего минимального по модулю числа: %i\n",min+1);
    // подсчитываем сумму модулей после минимального
    printf("Сумма чисел после минимального: %i\n",summArray(m,n,min));
    // заменяем отрицательные числа квадратами
    negateCorrect(m,n);
    printf("Массив после изменения отрицательных чисел:\n");
    showArray(m,n);
    // сортируем массив
    sortArray(m,n);
    printf("Сортированный массив:\n");
    showArray(m,n);

    return 0;
}

```