ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ  
Ордена Трудового Красного Знамени федеральное государственное   
бюджетное учреждение высшего образования

Московский технический университет связи и информатики

МТУСИ

Кафедра информационной безопасности

Лабораторная работа

Вариант 2

Выполнил студент группы БИН1703 ОТФ-2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бен Режеб Софиэн

Старший преподаватель кафедры ИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Барков

**Практикум №4**  
**Указатели и массивы**

**Цель практикума:** Овладеть навыками работы с указателями и массивами в языке программирвоания C

***Задание 1***

Линейный алгоритм функции из задания 1 практикума №1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода. Каждую процедуру оформить как функцию. Вычислительную часть алгоритма оформить как функцию с параметрами, передаваемыми по значению с использованием указателей на константные объекты, без возвращаемого значения. Результат вычисления вернуть через параметр-указатель. Прототип функции:

void f(const double \*x, double \*result) // Если функция имеет один параметр

void f(const double \*x, const double \*y, double \*result) // Если функция имеет два параметра

Другую часть алгоритма оформить как функцию void main(), вызывающую первую функцию нужное количество раз. Записать тексты функций файл с именем task1.с в следующем порядке: функция с параметрами, функция main. Cкомпилировать, скомпоновать и выполнить.

***Задание 2***

Написать функцию вывода массива целых чисел на экран. Функция получает указатель на первый элемент массива и его длину.

Прототип функции void writeArray(int \*arr, int n).

***Задание 3***

Написать функцию ввода массива целых чисел с клавиатуры. Функция получает указатель на первый элемент массива и его длину.

Прототип функции void readArray(int \*arr, int n).

***Задание 4***

Написать функцию сортировки массива с использованием алгоритма сортировки, указанном в индивидуальном задании. Функция должна получать указатель на первый элемент массива и количество элементов в массиве.

Прототип функции void sort(int \*arr, int n).

Функция не должна использовать функции консольного ввода-вывода.

***Задание 5***

Написать функцию, которая создаёт на стеке массив из N элементов (число N определяется константой в коде), выводит на экран N и с помощью разработанных ранее функций (задания 2-4) осуществляет ввод данных, вывод массива на экран, сортировку и повторный вывод отсортированного массива на экран

Прототип функции void sortStackArray(int \*arr, int n).

***Задание 6***

Разработать функцию main, демонстрирующую работу функций из заданий 1 и 5. Организовать меню и возможность многократной демонстрации заданий.

**Индивидуальные варианты заданий**  
Сортировка Шелла

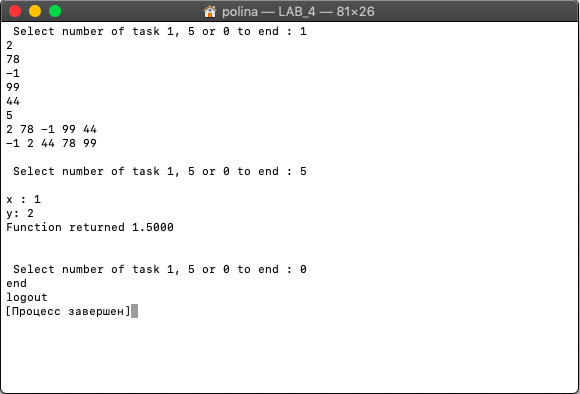


Рисунок 1 — Результат работы программы

Листинг 1 — Исходный файл LAB\_4.c

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

**void** function(**double** \*x, **double** \*y, **double** \*result)

{

\*result = (4-pow(\*x,2))/2;

}

**void** writeArray(**int** \*arr, **int** n)

{

**int** i;

**for** (i = 0; i < n; i++)

printf("%d ", arr[i]);

printf("\n");

}

**void** readArray(**int** \*arr, **int** n)

{

**int** i;

**for** (i = 0; i < n; ++i)

scanf("%d", arr + i);

}

**void** ShellSort(**int** \*arr,**int** n)

{

**int** i, j, step;

**int** tmp;

**for** (step = n / 2; step > 0; step /= 2)

**for** (i = step; i < n; i++)

{

tmp = arr[i];

**for** (j = i; j >= step; j -= step)

{

**if** (tmp < arr[j - step])

arr[j] = arr[j - step];

**else**

**break**;

}

arr[j] = tmp;

}

}

**void** sortStackArray(**int** \*arr, **int** n)

{

readArray(arr, n);

printf("%d\n", n);

writeArray(arr, n);

ShellSort(arr, n);

writeArray(arr, n);

}

**int** main()

{

**double** x, y, result;

**int** N = 5;

**int** arr[5];

**int** f = 1;

**for** (**int** i = 0; i < f; ++i)

{

++f;

printf("\n Select number of task 1, 5 or 0 to end : ");

**switch** (getchar())

{

**case** '1':

sortStackArray(arr, N);

getchar();

**continue**;

**case** '5':

printf("\n");

printf("x : ");

scanf("%lf", &x);

printf("y: ");

scanf("%lf", &y);

function(&x, &y, &result);

printf("Function returned %.4lf\n\n", result);

getchar();

**continue**;

**case** '0' :

printf("end\n");

f = f - 1;

**break**;

**default**:

getchar();

**continue**;

}

}

}