Министерство цифрового развития, связи и массовых коммуникаций

Российской Федерации

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение

высшего образования

Московский технический университет связи и информатики

(МТУСИ)

Кафедра «Информационная безопасность»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине

«Программирование в системах информационной безопасности»

на тему

## «Указатели и массивы»

Вариант №17

Выполнил:

студент группы БСУ1801

Неронов Ф.А.

Проверил:

старший преподаватель кафедры ИБ

Барков В. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2021

**Цель**

Овладеть навыками работы с указателями и массивами в языке программирования C.

Вариант №17

«Сортировка выбором»

**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.

Листинг 1 – Исходный код файла task1.c

#include <stdio.h>

#include <math.h>

#include <conio.h>

#include <stdlib.h>

void f(const double\* x, double\* result)

{

\*result = sqrt(pow((\*x) \* 3 + 2, 2) - 24 \* (\*x)) / (3 \* sqrt((\*x)) - (2 / sqrt((\*x))));

}

int main(void)

{

double x, result;

\_Bool b = 1;

while (b == 1)

{

printf("Menu:\n");

printf("1. Start\n");

printf("2. End\n");

printf("Select Task number: ");

switch (\_getch())

{

case '1':

{

printf("\nx = ");

scanf("%lf", &x);

f(&x, &result);

printf("f = %.4lf\n\n", result);

break;

}

case '2':

{

b = 0;

break;

}

default:

{

printf("\nInvalid value entered. Try again.\n");;

break;

}

}

}

system("pause");

return 0;

}

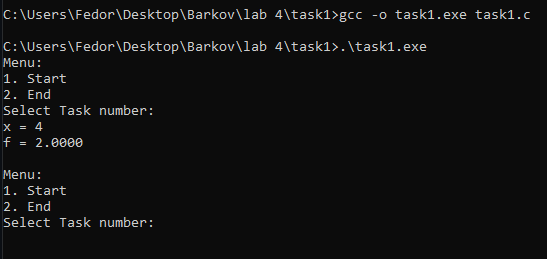


Рисунок 1 – Пример работы функции main в файле task1.c

**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).

Формат вывода: 1 2 3 4 5 6

Листинг 2 – Исходный код файла task2.c

#include <stdio.h>

void writeArray(int\* arr, int n)

{

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

{

printf("%i\t", \*(arr + i));

}

printf("\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).

Листинг 3 – Исходный код файла task3.c

#include <stdio.h>

void readArray(int\* arr, int n)

{

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

{

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

scanf("%i", &arr[i]);

}

}

**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.

Листинг 4 – Исходный код файла task4.c

void sort(int\* arr, int n)

{

int tmp = 0;

for (int i = 0; i < n - 1; i++)

{

int minPosition = i;

for (int j = i; j < n; j++)

{

if (arr[minPosition] > arr[j])

minPosition = j;

}

if (i != minPosition)

{

tmp = arr[i];

arr[i] = arr[minPosition];

arr[minPosition] = tmp;

}

}

}

**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).

Формат вывода (первая строка – количество элементов массива):

5

5 1 9 7 8

1 5 7 8 9

Листинг 5 – Исходный код файла task5.c

#include <stdio.h>

#include "func.h"

void sortStackArray(int\* arr, int n)

{

n = 5;

readArray(arr, n);

printf("%i", n);

printf("\n");

writeArray(arr, n);

sort(arr, n);

writeArray(arr, n);

}

**Задание 6**

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

Листинг 6 – Исходный код файла func.h

#ifndef FUNC\_H

#define FUNC\_H

extern double x, result;

extern int n;

//Task 1

void f(const double\* x, double\* result);

//Task 2

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

//Task 3

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

//Task 4

void sort(int\* arr, int n);

//Task 5

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

#endif

Листинг 7 – Исходный код файла task6.c

#include <stdio.h>

#include <math.h>

#include <conio.h>

#include <stdlib.h>

#include "func.h"

int main(void)

{

\_Bool b = 1;

while (b == 1)

{

printf("Menu:\n");

printf("1. Task 1\n");

printf("5. Task 5\n");

printf("6. Exit \n");

printf("Select Task number: ");

switch (\_getch())

{

case '1':

{

printf("Task 1:\n ");

double x, result;

printf("x = ");

scanf("%lf", &x);

f(&x, &result);

printf("f = %.4lf\n\n", result);

break;

}

case '5':

{

printf("Task 5:\n ");

int n;

int arr[n];

sortStackArray(arr, n);

break;

}

case '6':

{

b = 0;

break;

}

default:

{

printf("\nInvalid value entered. Try again.\n");;

break;

}

}

}

system("pause");

return 0;

}

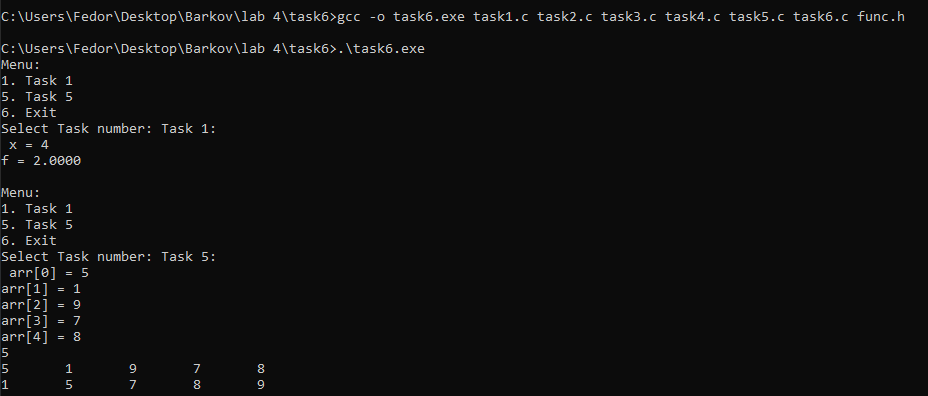


Рисунок 2 – Пример работы функции main

<https://github.com/BSU1801/Barkov_lab4.git>

**Вывод**

В данной работе были разобраны аспекты работы с указателями и массивами. Так же были изучены различия между указателями и проведена работа с арифметикой указателей при работе с массивами.