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

Выполнил:

студент группы БСУ1801

Спирин В.В.

Проверил:

старший преподаватель кафедры ИБ

Барков В.В.

Москва, 2021

# Цель работы

Овладеть навыками работы с указателями и массивами в языке программирования C

# Задание

По номеру Вашего варианта выбрать задачу, решаемую в этой лабораторной работе, и выполнить для нее следующие задания.

**Индивидуальное задание**

9 вариант

**Общее Задание**

**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).

Формат вывода: 1 2 3 4 5 6

**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).

Формат вывода (первая строка – количество элементов массива):

5

5 1 9 7 8

1 5 7 8 9

**Задание 6**

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

# Выполнение

Листинг 1 – программный код файла task1.c

#include <math.h>

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

{

\*result = pow(cos(\*x) - cos(\*y), 2) - pow(sin(\*x) - sin(\*y), 2);

}

Листинг 2 – программный код файла task2.c

#include <stdio.h>

void writeArray(int\* arr, int n)

{

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

{

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

}

printf("\n");

}

Листинг 3 – программный код файла task3.c

#include <stdio.h>

void readArray(int\* arr, int n)

{

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

{

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

}

}

Листинг 4 – программный код файла task4.c

void Sort(int\* arr, int n) {

int a = 0;

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

for (int j = n - 1; j > i; j--) {

if (arr[j] < arr[j - 1]) {

a = arr[j - 1];

arr[j - 1] = arr[j];

arr[j] = a;

}

}

}

}

Листинг 5 – программный код файла task5.c

#include <stdio.h>

#include "Header.h"

void sortStackArray(int\* arr, int n)

{

printf(" из %d \n", n);

readArray(arr, n);

writeArray(arr, n);

Sort(arr, n);

writeArray(arr, n);

}

Листинг 6 – программный код файла header.h

#pragma once

#define N 5

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

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

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

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

Листинг 7 – программный код файла main.c

#include <locale.h>

#include <stdio.h>

#include "header.h"

#include <stdbool.h>

void main() {

int arr[N];

const double x = 5;

const double y = 8.2537;

double result = 0;

\_Bool close\_flag = true;

setlocale(LC\_CTYPE, "rus");

while (close\_flag) {

printf("1. Задание 1 \n2. Задание 2 \n3. выход \n");

switch (getch()) {

case '1':

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

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

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

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

printf("x=");

scanf\_s("%lf", &x);

printf("y=");

scanf\_s("%lf", &y);

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

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

break;

case '2':

printf("введите массив\n");

sortStackArray(arr, N);

break;

case '3':

close\_flag = 0;

break;

}

}

}

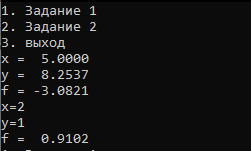


Рисунок 1 – Результат работы с программой (Задание 1)

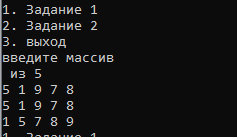


Рисунок 2 – Результат работы с программой (Задание 2)