**Ордена Трудового Красного Знамени**

Федеральное Государственное

бюджетное образовательное учреждение

высшего образования

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

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СВЯЗИ И ИНФОРМАТИКИ**

Кафедра Информационной Безопасности

**ЛАБОРАТОРНАЯ РАБОТА**

**№ 4**

по дисциплине «Объектно-ориентированное программирование система защиты информации»

на тему:

**«Производные типы: указатели, массивы, строки, структуры, объединения. Статическое и динамическое выделение памяти»**

Вариант №16

Выполнила: студ. гр. БПЗ1501

Романюк И. Ф.

Проверил: ассисент кафедры ИБ

Барков В. В.

Москва 2018

**Цель работы**

Овладеть навыками работы с производными типами данных в языке C, научиться динамически выделять память.

**Выполнение**

Задание 1

Написать функцию, которая в качестве параметра получает указатель на целое число и увеличивает его значение на единицу. Функция должна осуществлять проверку переданного указателя на нуль. Перед началом выполнения и перед выходом функция выводит адрес и значение параметра. Прототип функции void increment(int \*n). Формат вывода: адрес значение адрес значение.

Исходный код:

#include "Header.h"

void increment(int \*n)

{

printf("\n Адрес указателя: %p\n Значение указателя: %p\n Значение переменной по указателю: %d\n", &n, n, \*n);

++\*n;

printf("\n Адрес указателя: %p\n Значение указателя: %p\n Значение переменной по указателю: %d\n\n", &n, n, \*n);

}

Задание 2

Написать функцию, которая создаёт на стеке целое число, осуществляет ввод, выводит на экран адрес переменной и её значение. Далее вызывает функцию из задания 1, а затем повторно выводит адрес и значение переменной. Прототип функции void incrementStackVariable(). Формат вывода: адрес значение адрес значение адрес значение адрес значение.

Исходный код:

#include "Header.h"

void incrementStackVariable()

{

int x;

printf("x = "); scanf("%d", &x);

printf("Адрес переменной: %p\nЗначение переменной: %d\n", &x, x);

increment(&x);

printf("Адрес переменной: %p\nЗначение переменной: %d\n", &x, x);

}

Задание 3

Написать функцию, которая создаёт в динамической памяти целое число, осуществляет ввод, выводит на экран адрес переменной и её значение. Далее вызывает функцию из задания 1, а затем повторно выводит адрес и значение переменной. Перед выходом из функции необходимо освободить ранее выделенную память. Прототип функции void incrementHeapVariable(). Формат вывода: адрес значение адрес значение адрес значение адрес значение.

Исходный код:

#include "Header.h"

void incrementHeapVariable()

{

int \*x = (int \*)malloc(sizeof(int));

printf("x = "); scanf("%d", x);

printf("Адрес переменной: %p\nЗначение переменной: %d\n", x, \*x);

increment(x);

printf("Адрес переменной: %p\nЗначение переменной: %d\n", x, \*x);

free(x);

}

Задание 4

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

Исходный код:

#include "Header.h"

void writeArray(int \*arr, int n)

{

printf("\nЗадание 4:\n");

for (int \*i = arr; (i - arr) < n; i++)

{

printf("x[%d] = %d\n", i - arr, \*i);

}

}

Задание 5

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).

Исходный код:

#include "Header.h"

void readArray(int \*arr, int n)

{

printf("\nЗадание 5:\n");

for (int \*i = arr; (i - arr) < n; i++)

{

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

scanf("%d", i);

}

}

Задание 6

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

Исходный код:

#include "Header.h"

void sort(int \*arr, int n)

{

\_Bool end = 1;

int a;

while (end)

{

end = 0;

for (int \*i = arr; (i - arr) < n - 1; i++)

{

if (\*(i) > \*(i + 1))

{

a = \*(i);

\*(i) = \*(i + 1);

\*(i + 1) = a;

end = 1;

}

}

}

}

Задание 7

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 4-6) осуществляет ввод данных, вывод массива на экран, сортировку и повторный вывод отсортированного массива на экран Прототип функции void sortStackArray(int \*arr, int n).

Исходный код:

#include "Header.h"

void sortStackArray()

{

int n = 10;

int x[10];

readArray(x, n);

writeArray(x, n);

sort(x, n);

writeArray(x, n);

}

Задание 8

Написать функцию, которая создаёт в динамической памяти из N элементов (число N вводится пользователем с клавиатуры), выводиn n на экран и с помощью разработанных ранее функций (задания 4-6) осуществляет ввод данных, вывод массива на экран, сортировку и повторный 20 вывод отсортированного массива на экран. Перед выходом из функции необходимо освободить выделенную ранее память. Прототип функции void sortHeapArray(int \*arr, int n).

Исходный код:

#include "Header.h"

void sortHeapArray()

{

int n = 10;

printf("n = "); scanf("%d", &n);

int \*x = (int \*)malloc(n \* sizeof(int));

readArray(x, n);

writeArray(x, n);

sort(x, n);

writeArray(x, n);

free(x);

}

Задание 9

Разработать функцию main, демонстрирующую работу функций из заданий 2,3,7,8. Организовать меню и возможность многократной демонстрации заданий.

Исходный код:

#include "Header.h"

int main()

{

setlocale(LC\_ALL, "Rus");

while (1)

{

system("cls");

printf("Введите номер действия:\n2 - Задание 2\n3 - Задание 3\n7 - Задание 7\n8 - Задание 8\n0 - Выход\n\n");

switch (getch())

{

case '2':

system("cls");

printf("Задание 2:\n");

incrementStackVariable();

getch();

break;

case '3':

system("cls");

printf("Задание 3:\n");

incrementHeapVariable();

getch();

break;

case '7':

system("cls");

printf("Задание 7:\n");

sortStackArray();

getch();

break;

case '8':

system("cls");

printf("Задание 8:\n");

sortHeapArray();

getch();

break;

case '0':

return 0;

break;

default:

system("cls");

printf("Куда жмешь?!\n");

getch();

break;

}

}

}

Header.h

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "locale.h"

void increment(int \*n);

void incrementStackVariable();

void incrementHeapVariable();

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

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

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

void sortStackArray();

void sortHeapArray();

Результаты:  






 