**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Ордена трудового красного знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ**

КАФЕДРА РАДИООБОРУДОВАНИЯ И СХЕМОТЕХНИКИ

**Лабораторная работа № 4**

по дисциплине «Объектно-ориентированное программирование систем защиты информации»:

на тему:

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

Вариант №14

Выполнил:

Студент группы БПЗ1501

Поляев Антон

Москва, 2018

# Цель работы

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

# Результаты

## Задание №1

#include "Header.h"

void task1(int \*x)

{

printf("\n Задание 1:\n Адрес указателя: %p\n Значение указателя: %p\n Значение переменной по указателю: %d\n", &x, x, \*x);

++\*x;

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

}

## }Задание №2

#include "Header.h"

void task2()

{

int x;

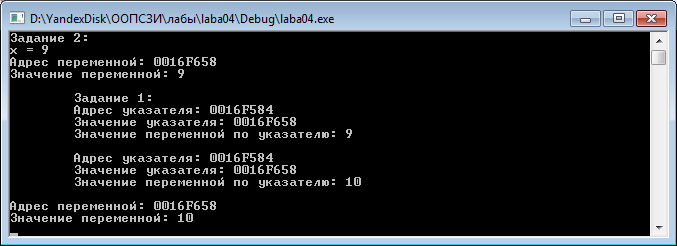
printf("x = "); scanf("%d", &x);

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

task1(&x);

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

}



## Задание №3

#include "Header.h"

void task3()

{

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

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

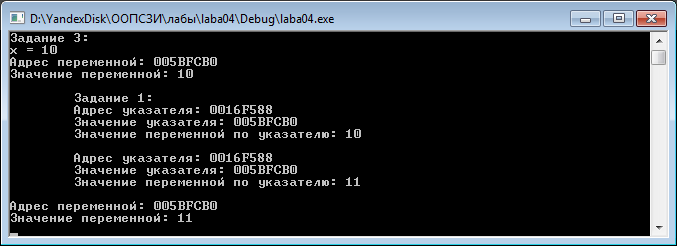
printf("Адрес переменной: %p\nЗначение переменной: %d\n", x, \*x);

task1(x);

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

free(x);

}



## }Задание №4

#include "Header.h"

void task4(int \*x, int n)

{

printf("\nЗадание 4:\n");

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

{

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

}

}

## Задание №5

#include "Header.h"

void task5(int \*x, int n)

{

printf("\nЗадание 5:\n");

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

{

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

scanf("%d", i);

}

}

## Задание №6

#include <iostream>

#include <ctime>

#include <iomanip>

using namespace std;

void insertionSort(int \*, int); // прототип функции сортировки вставками

int main(int argc, char\* argv[])

{

srand(time(NULL));

setlocale(LC\_ALL, "rus");

cout << "Введите размер массива: ";

int size\_array; // длинна массива

cin >> size\_array;

int \*sorted\_array = new int[size\_array]; // одномерный динамический массив

for (int counter = 0; counter < size\_array; counter++)

{

sorted\_array[counter] = rand() % 100; // заполняем массив случайными числами

cout << setw(2) << sorted\_array[counter] << " "; // вывод массива на экран

}

cout << "n";

insertionSort(sorted\_array, size\_array); // вызов функции сортировки вставками

for (int counter = 0; counter < size\_array; counter++)

{

cout << setw(2) << sorted\_array[counter] << " "; // печать отсортированного массива

}

cout << "n";

delete[] sorted\_array; // высвобождаем память

system("pause");

return 0;

}

void insertionSort(int \*arrayPtr, int length) // сортировка вставками

{

int temp, // временная переменная для хранения значения элемента сортируемого массива

item; // индекс предыдущего элемента

for (int counter = 1; counter < length; counter++)

{

temp = arrayPtr[counter]; // инициализируем временную переменную текущим значением элемента массива

item = counter - 1; // запоминаем индекс предыдущего элемента массива

while (item >= 0 && arrayPtr[item] > temp) // пока индекс не равен 0 и предыдущий элемент массива больше текущего

{

arrayPtr[item + 1] = arrayPtr[item]; // перестановка элементов массива

arrayPtr[item] = temp;

item--;

}

}

}

endSort = 1;

}

}

}

}

Задание №7

#include "Header.h"

void task7()

{

int n = 10;

int x[10];

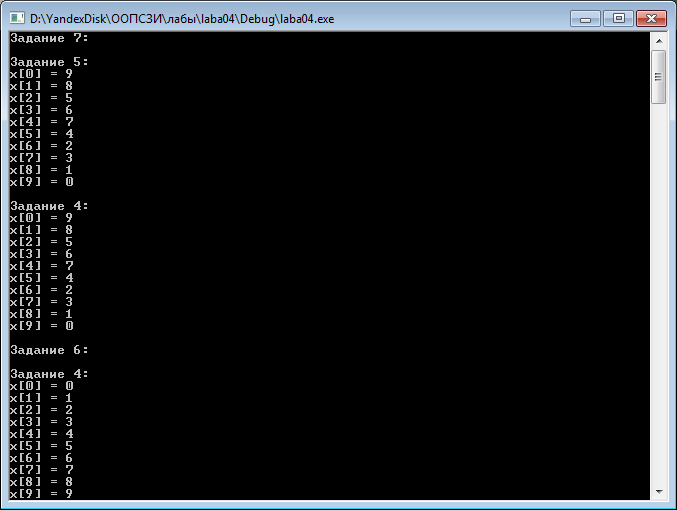
task5(x, n);

task4(x, n);

task6(x, n);

task4(x, n);

}



## Задание №8

#include "Header.h"

void task8()

{

int n = 10;

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

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

task5(x, n);

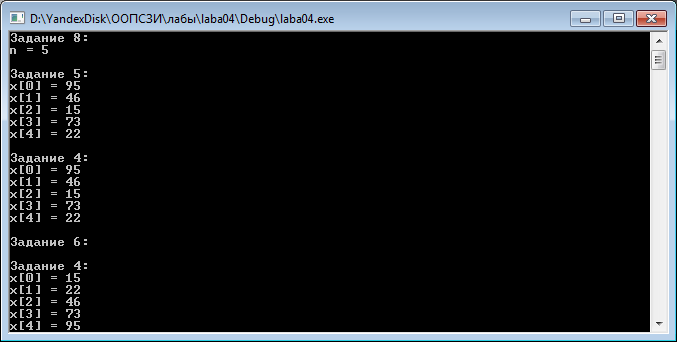
task4(x, n);

task6(x, n);

task4(x, n);

free(x);

}



## Задание №9

#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");

task2();

\_getch();

break;

case '3':

system("cls");

printf("Задание 3:\n");

task3();

\_getch();

break;

case '7':

system("cls");

printf("Задание 7:\n");

task7();

\_getch();

break;

case '8':

system("cls");

printf("Задание 8:\n");

task8();

\_getch();

break;

case '0':

return 0;

break;

default:

system("cls");

printf("Неверный ввод\n");

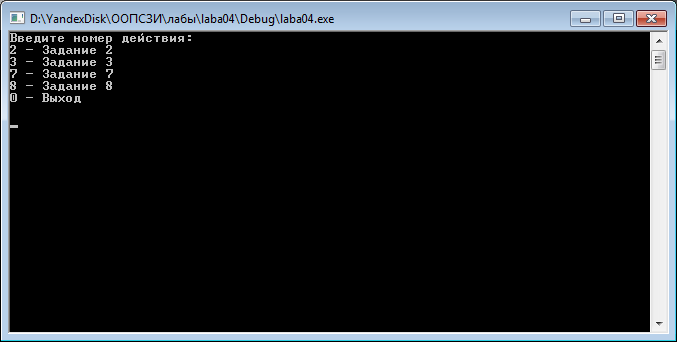
\_getch();

break;

}

}

}



## Header.h

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "locale.h"

void task1(int \*x);

void task2();

void task3();

void task4(int \*x, int n);

void task5(int \*x, int n);

void task6(int \*x, int n);

void task7();

void task8();