**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

**ЛАБОРАТОРНАЯ РАБОТА №4**

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

Выполнил:

Студент группы БПЗ1602

Кузмин Д.А.

Проверил:

Ассистент кафедры ИБ

Барков В.В.

Москва, 2019

# 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;

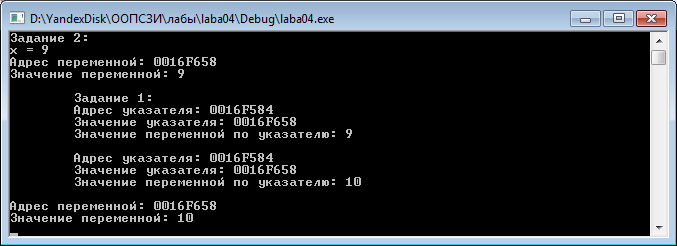
printf("x = "); scanf("%d", &x);

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

task1(&x);

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

}



1. Результат

## Задание №3

#include "Header.h"

void task3()

{

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

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

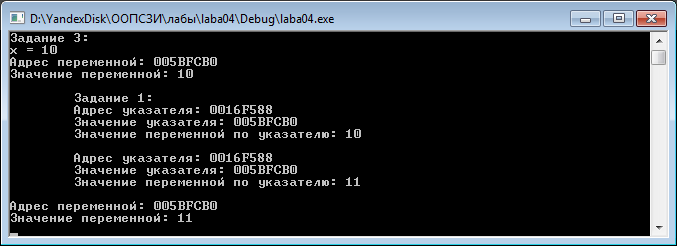
printf("Адрес переменной: %p\nЗначение переменной: %d\n", x, \*x);

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 "Header.h"

void task6(int \*x, int n)

{

\_Bool endSort = 1;

int obmen;

printf("\nЗадание 6:\n");

while (endSort)

{

endSort = 0;

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

{

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

{

obmen = \*(i);

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

\*(i + 1) = obmen;

endSort = 1;

}

}

}

}

## Задание №7

#include "Header.h"

void task7()

{

int n = 10;

int x[10];

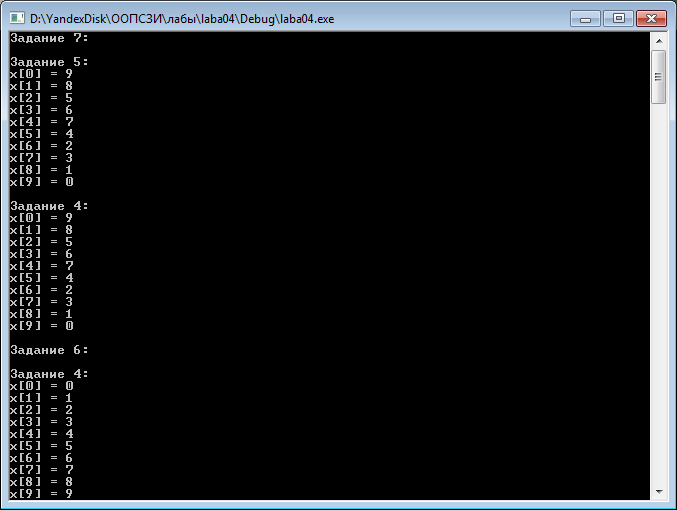
task5(x, n);

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);

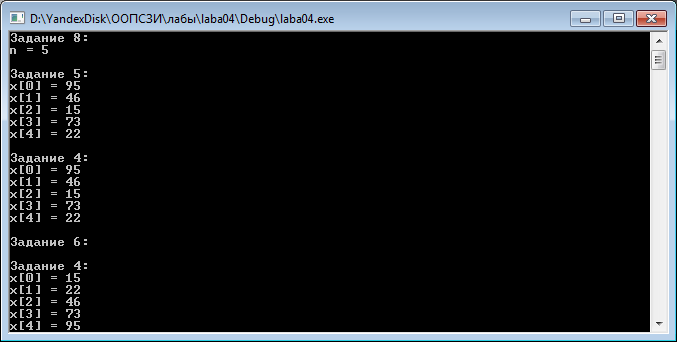
task4(x, n);

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");

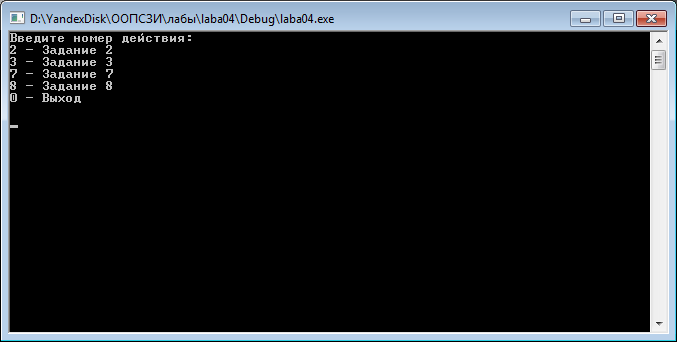
\_getch();

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();