**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

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

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СВЯЗИ И ИНФОРМАТИКИ**

Кафедра «Информационная безопасность»

**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 №10

Выполнил: студ. гр. БПЗ1501

Лисеенко Я.А.

Проверил: ассистент кафедры ИБ

Барков В.В.

Москва 2018 г

**Цель:** овладеть навыками работы с производными типами данных в языке C, научиться динамически выделять память.

**Выполнение:**

**Задание 1:**

void increment(int \*n) {

printf("task1 \n");

if ((\*n) != 0) {

printf("%p %d \n", n, \*n);

(\*n)++;

printf("%p %d \n", n, \*n);

}

}

**Задание 2:**

void incrementStackVariable() {

printf("task2 \n");

int x;

printf("x = ");

scanf("%d", &x);

printf("%p %d \n", &x, x);

increment(&x);

printf("%p %d \n", &x, x);

}

**Задание 3:**

void incrementHeapVariable() {

printf("task3 \n");

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

printf("p = ");

scanf("%d", p);

printf("%p %d \n", p, \*p);

increment(p);

printf("%p %d \n", p, \*p);

free(p);

}

**Задание 4:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

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

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

printf("arr[%d] = %d \n", i + 1, arr[i]);

}

}

**Задание 5:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

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

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

printf("arr[%d] = ", i + 1);

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

}

}

**Задание 6:**

void sort(int \*arr, int left, int right) {

//Быстрая сортировка

int pivot;

int l\_hold = left;

int r\_hold = right;

pivot = arr[left];

while (left < right)

{

while ((arr[right] >= pivot) && (left < right))

right--;

if (left != right)

{

arr[left] = arr[right];

left++;

}

while ((arr[left] <= pivot) && (left < right))

left++;

if (left != right)

{

arr[right] = arr[left];

right--;

}

}

arr[left] = pivot;

pivot = left;

left = l\_hold;

right = r\_hold;

if (left < pivot)

sort(arr, left, pivot - 1);

if (right > pivot)

sort(arr, pivot + 1, right);

}

**Задание 7:**

void sortStackArray() {

const int n = 5;

int l = 0, r = n - 1;

int arr[5];

readArray(arr, n);

printf("n = %d \n", n);

writeArray(arr, n);

sort(arr, l, r);

printf("Sortirovka: \n");

writeArray(arr, n);

}

**Задание 8:**

void sortHeapArray() {

int n, l, r;

printf("n = ");

scanf("%d", &n);

l = 0; r = n - 1;

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

readArray(arr, n);

printf("n = %d \n", n);

writeArray(arr, n);

printf("Sortirovka: \n");

sort(arr, l, r);

writeArray(arr, n);

}

**Задание 9:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include "Header.h"

int main() {

int n;

for (int i = 0; i != -1;) {

system("cls");

printf("1 to task 2 \n");

printf("2 to task 3 \n");

printf("3 to task 7 \n");

printf("4 to task 8 \n");

printf("5 to Exit \n \n");

printf("Choose Your Destiny");

switch (\_getch()) {

case '1':

system("cls");

printf("task 2 \n");

incrementStackVariable();

\_getch();

break;

case '2':

system("cls");

printf("task 3 \n");

incrementHeapVariable();

\_getch();

break;

case '3':

system("cls");

printf("task 7 \n");

sortStackArray();

\_getch();

break;

case '4':

system("cls");

printf("task 8 \n");

sortHeapArray();

\_getch();

break;

case '5':

system("cls");

printf("Exit");

i = -1;

break;

default:

break;

}

}

\_getch();

return 0;

}

**Результат:**

 



