Самостоятельная работа №11.

**Тема: Структуры и объединения в языке С**

Задание №9.1.

**Постановка задачи:** Напишите программу, в которой реализуется замкнутая цепочка из экземпляров структуры: каждый экземпляр ссылается на следующий, а последний экземпляр ссылается на первый.

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

struct one

{

struct one \* two;

};

struct two

{

struct two \* tree;

};

struct tree

{

struct tree \* four;

};

struct four

{

struct four \* five;

};

struct five

{

struct five \* one;

};

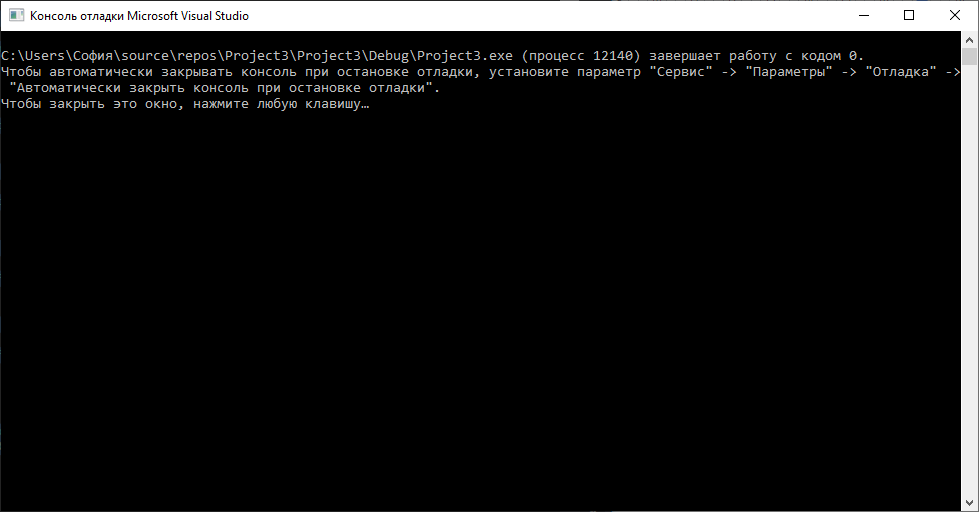
int main()

{

struct one;

return 0;

}

**Результат:**

Задание №9.2.

**Постановка задачи:** Напишите программу для реализации операций с комплексными числами, представленными в тригонометрической форме (то есть в виде z = r·exp(iφ). Модуль

r и аргумент φ реализуются в виде полей структуры.

**Код программы:**

**Результат:**

Задание №9.3.

**Постановка задачи:** Напишите программу, в которой с помощью структур реализуется бинарное дерево: каждый экземпляр структуры элемента дерева (за исключением экземпляров последнего уровня - листьев дерева) содержит ссылку на два других

экземпляра (поддерево или лист). Реализовать:

* заполнение дерева вручную в программном коде (простой вариант);
* распечатка дерева используя рекурсию с поиском в глубину 1, либо используя стек вместо рекурсии (Задача 9.4).

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

typedef struct Tree

{

int a;

struct Tree\* next;

struct Tree\* prev;

}tree;

void ctree(tree \*r, int lev)

{

tree \*left, \*right;

r->a = lev;

if (lev)

{

left = malloc(sizeof(tree));

right = malloc(sizeof(tree));

r->next = left;

r->prev = right;

lev--;

ctree(left, lev);

ctree(right, lev);

}

else

{

r->next = NULL;

r->prev = NULL;

}

}

void cleen(tree\*r)

{

if (r->next != NULL)

cleen(r->next);

if (r->prev != NULL)

cleen(r->prev);

free(r);

}

void write(tree\*r)

{

printf("%2d ", r->a);

if (r->next != NULL)

write(r->next);

if (r->prev != NULL)

write(r->prev);

}

int main()

{

tree\*r=malloc(sizeof(tree));

ctree(r, 3);

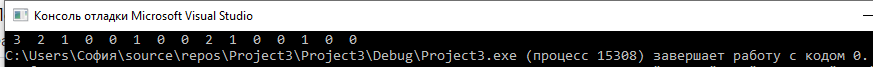
write(r);

cleen(r);

return 0;

}

**Результат:**



Задание №9.4.

**Постановка задачи:** Реализовать структуру данных стэк (struct Stack) на основе двунаправленного списка элементов (nodes). Каждый элемент (struct Node) содержит некоторые полезные данные и указатели на предыдущий и следующий элементы в

стеке. Реализовать для стека следующие операции:

* push – помещение элемента в стек на его вершину;
* pop – удаление элемента элемента из вершины стека с возвратом этого
* удаленного элемента (node) из функции pop;
* peek – возврат элемента (node) вершины стека (чтение вершины) из функции peek без удаления его из стека;
* is\_empty – проверка того, что стек является пустым (не содержит элементов вообще).

**Код программы:**

**Результат:**

Задание №9.5.

**Постановка задачи:** Реализовать структуру (struct) с вложенным поименованным объединением (union) и дополнительным целочисленным полем внутри данной структуры, в котором по условному номеру хранится информация о том, какие именно данные записаны в основное поле типа union. Внутри данного вложенного объединения определить два поля типа char и int. Создать и заполнить динамический массив таких структур целыми и символьными данными, заполняя вспомогательное целое поле, для сохранения информации о хранимом в каждом объединении типе данных (char/int). Реализовать распечатку данных массива таких структур в консоль.

**Код программы:**

**Результат:**