Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра микро- и наноэлектроники

**Лабораторная работа №3**

**«Динамическая структура СТЕК»**

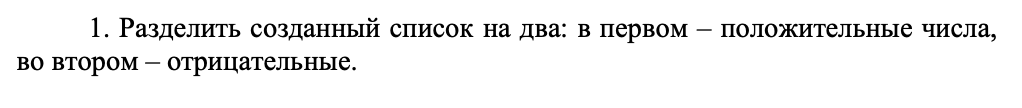
**Вариант 1**

|  |  |
| --- | --- |
| Выполнил: | студент группы 142701  Ахремчик Никита Андреевич |
| Проверил: |  |

Минск 2022

**Задачи:** изучить алгоритмы работы с динамическими структурами данных в виде стека.

**Индивидуальное задание**



Текст программы:

#include <iostream>

#include <stdlib.h>

#include <time.h>

using namespace std;

struct stack{

int data;

stack \***next**;

};

typedef void (\*func)(stack \*\*, int);

bool empty(stack \*st){

if (st == nullptr) **return** **true**;

else **return** **false**;

}

void output(stack \*st){

if (empty(st))

printf("Стек пустой!");

else{

printf("Содержание стека: ");

**while**(st){

printf("%i ", st->data);

st = st->**next**;

}

}

puts("");

}

void add(stack \*\*top, int data){

stack \*elem = new stack();

elem->data = data;

if (top == NULL)

\*top = elem;

else{

elem->**next** = \*top;

\*top = elem;

}

}

void del(stack \*\*top, int elem=NULL){

stack \*start = \*top;

stack \*prev = NULL;

**while** (start != NULL){

if (start->data == elem){

\*top = start->**next**;

free(start); // очищаем память

start->data = NULL;

start->**next** = NULL;

}

else{

prev->**next** = start->**next**;

free(start); // очищаем память

start->data = NULL;

start->**next** = NULL;

}

prev = start;

start = start->**next**;

};

}

void sort\_address(stack \*\*st){

stack \*start = NULL, \***end**, \*temp;

if ((\*st)->**next**->**next** == NULL) **return**;

**do** {

**for** (**end**=\*st; **end**->**next**->**next** != start; **end**=**end**->**next**){

if (**end**->**next**->data > **end**->**next**->**next**->data){

temp = **end**->**next**->**next**;

**end**->**next**->**next** = temp->**next**;

temp->**next** = **end**->**next**;

**end**->**next** = temp;

}

}

start= **end**->**next**;

} **while** ((\*st)->**next**->**next** != start);

}

void sort\_data(stack \*st){

stack \*start = NULL, \***end**;

int temp;

**do**{

**for** (**end**=st; **end**->**next** != start; **end**=**end**->**next**){

if (**end**->data > **end**->**next**->data){

temp = **end**->data;

**end**->data = **end**->**next**->data;

**end**->**next**->data = temp;

}

}

start = **end**;

} **while**(st->**next** != start);

}

void pop(stack \*s, int &size, int pos){

**for**(int i=pos-1; i<size-1; i++) s[i] = s[i+1];

size--;

}

void divide(stack \*\*st, stack \*\*st1, stack \*\*st2, func add){

stack \*start = NULL, \***end**;

**end** = \*st;

stack \*temp;

**do**{

if (**end**->data >= 0){

add(st1, **end**->data);

}

else{

add(st2, **end**->data);

}

**end** = **end**->**next**;

}**while** (**end** != start);

}

int main() {

srand(time(NULL));

stack \*st = NULL, \*st1 = NULL, \*st2 = NULL;

bool flag=**false**;

**while**(!flag){

int num, data;

printf("ВЫБЕРИТЕ РЕЖИМ\n1 - ЗАПОЛНИТЬ СТЕК\n2 - ОТСОРТИРОВАТЬ ПО АДРЕСУ\n3 - ОТСОРТИРОВАТЬ ПО ДАННЫМ\n4 - РАЗДЕЛИТЬ СТЕК\n5 - ПРОСМОТР СТЕКА\n0 - ВЫХОД\nВЫБЕРИТЕ ЗНАЧЕНИЕ: ");

int n;

cin >> n;

switch(n){

**case** 1:{

delete[] st;

delete[] st1;

delete[] st2;

printf("Длина стека: ");

cin >> num;

**for** (int i=0; i<num; i++){

data = rand() % 100 - 50;

add(&st, data);

}

**break**;

}

**case** 2:{

sort\_address(&st);

output(st);

**break**;

}

**case** 3:{

sort\_data(st);

output(st);

**break**;

}

**case** 4:{

divide(&st, &st1, &st2, &add);

output(st1);

output(st2);

del(&st);

**break**;

}

**case** 5:{

output(st);

**break**;

}

**case** 0:{

flag = **true**;

**break**;

}

}

}

delete[] st;

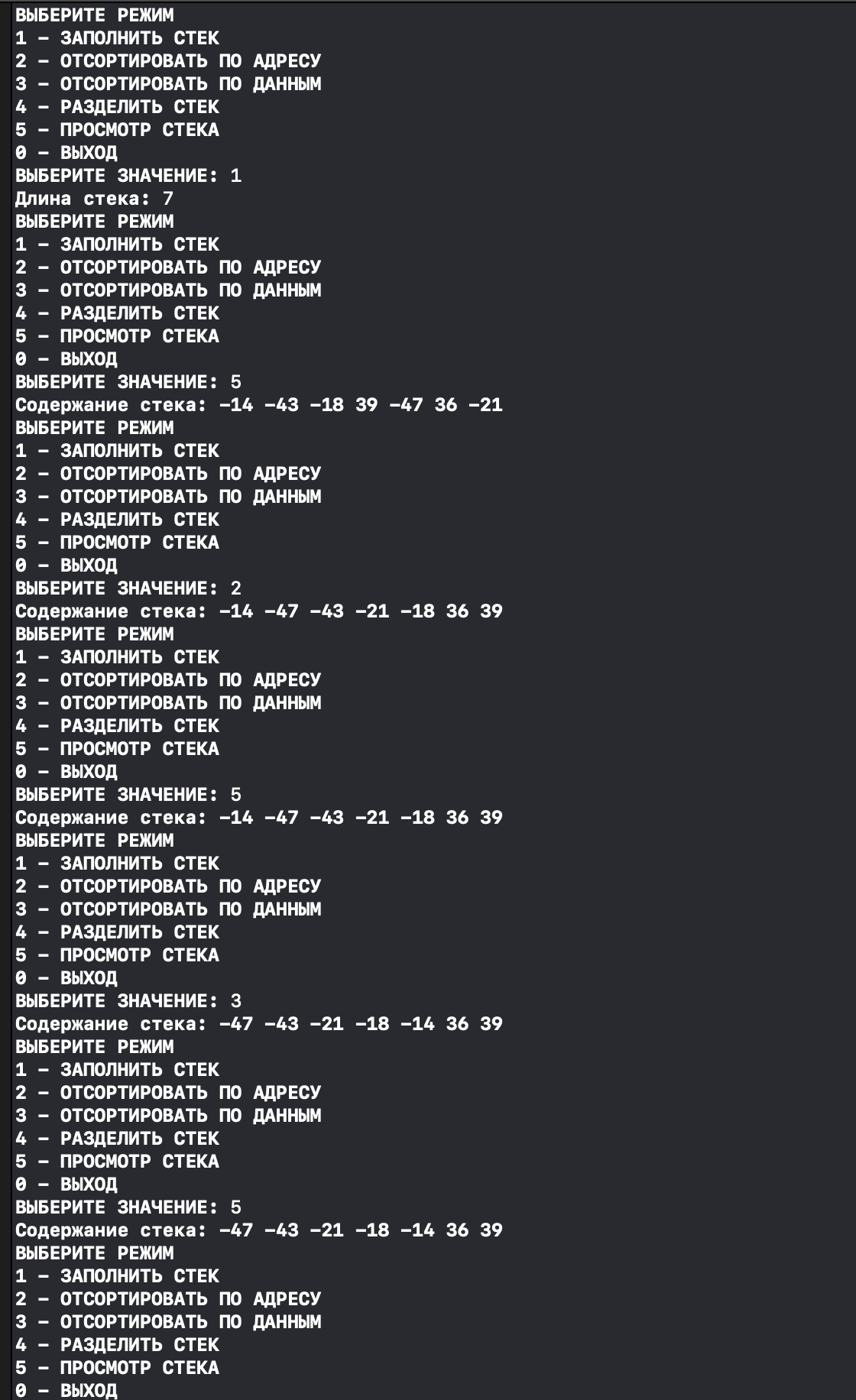
delete[] st1;

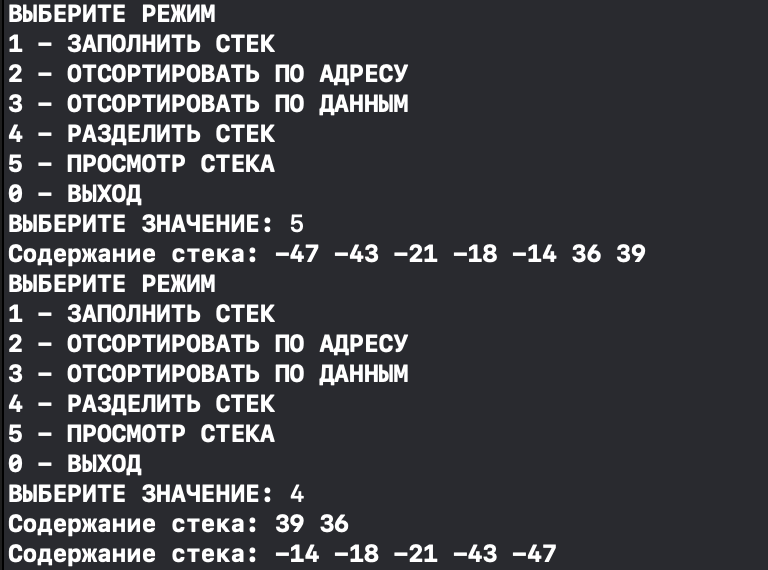
delete[] st2;

**return** 0;

}

**Результат вывода программы:**





**Выводы:** Проведя данную лабораторную работу, я научился работать с использованием динамических структур данных СТЕК в С++. Также закрепил знания по прошлым темам.