ГУО “БГУИР”

Факультет информационных технологий и управления

Кафедра систем управления

Отчет по

Лабораторной работе №2

Динамическая структура СТЕК

Подготовил:

Студент гр.222401

Рабинчук И.М.

Проверила:

Семижон Е.А.

Минск 2022

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

Написать программу по созданию, добавлению, просмотру и решению поставленной задачи (в рассмотренных примерах это действие отсутствует) для однонаправленного линейного списка типа СТЕК. Реализовать сортировку стека двумя рассмотренными выше методами.

Решение поставленной задачи представить в виде блок-схемы.

Во всех заданиях создать список из положительных и отрицательных случайных целых чисел.

Вариант №9. В созданном списке определить количество и удалить все элементы, находящиеся между минимальным и максимальным элементами.

Код:

#include <iostream>

#include <conio.h>

#include <stdlib.h>

#include <ctime>

**using** **namespace** std**;**

struct Stack

**{**

int info**;**

Stack **\***next**;**

**};**

Stack**\*** input**(**Stack **\***b**,** int a**)** //Функция создания или добавления элементов в стек

**{**

Stack **\***t **=** **new** Stack**;**

t**->**info**=**a**;**

t**->**next**=**b**;**

**return** t**;**

**}**

void view**(**Stack **\***b**)**

**{**

Stack **\***t **=** b**;**

**while** **(**t **!=** **NULL)**

**{**

cout **<<** t**->**info **<<** endl**;**

t **=** t**->**next**;**

**}**

**}**

Stack**\*** delete\_stack**(**Stack **\***b**)**

**{**

Stack**\***t**;**

**while** **(**b **!=** **NULL)**

**{**

t **=** b**;**

b **=** b**->**next**;**

**delete** t**;**

**}**

**return** b**;**

**}**

Stack**\*** out\_stack**(**Stack **\***b**,** int **\***out**)**

**{**

Stack**\*** t **=** b**;**

**\***out **=** b**->**info**;**

b **=** b**->**next**;**

**delete** t**;**

**return** b**;**

**}**

Stack**\*** Max**(**Stack **\***b**)**

**{**

Stack **\***max **=** b**;**

**while** **(**b**)**

**{**

**if** **(**b**->**info **>** max**->**info**)**

max **=** b**;**

**if** **(**b**->**info **==** max**->**info**)**

max **=** b**;** //???

b **=** b**->**next**;**

**}**

**return** max**;**

**}**

Stack**\*** Min**(**Stack**\***b**)**

**{**

Stack**\*** min **=** b**;**

**while** **(**b**)**

**{**

**if** **(**b**->**info **<** min**->**info**)**

min **=** b**;**

**if** **(**b**->**info **==** min**->**info**)**

min **=** b**;** //???

b **=** b**->**next**;**

**}**

**return** min**;**

**}**

Stack**\*** Work**(**Stack**\*** b**)**

**{**

**if** **(**b **==** **NULL)**

**return** b**;**

Stack**\*** max **=** Max**(**b**);**

Stack**\*** min **=** Min**(**b**);**

**if** **(**max **==** min**)**

**return** b**;**

**if** **(**min **==** max**)**

**return** b**;**

**if** **(**max**->**next **==** min**)**

**return** b**;**

**if** **(**min**->**next **==** max**)**

**return** b**;**

b **=** input**(**b**,** 9999**);**

Stack **\***p**=**b**;**

Stack **\***t **=** p**->**next**;**

int del\_numbs **=** 0**;**

**while** **(**t **!=** **NULL)**

**{**

**if** **(**t **==** max**)**

**{**

Stack**\*** tmp\_1 **=** t**->**next**;**

**while** **(**tmp\_1 **!=** min**)**

**{**

Stack**\*** tmp\_2 **=** tmp\_1**;**

tmp\_1 **=** tmp\_1**->**next**;**

**delete** tmp\_2**;**

del\_numbs**++;**

**}**

t**->**next **=** min**;**

t **=** b**;** //Удаление вершины

b **=** b**->**next**;**

**delete** t**;**

cout **<<** "Кол-во удаленных элементов: " **<<** del\_numbs **<<** endl**;**

**return** b**;**

**}**

**if** **(**t **==** min**)**

**{**

Stack**\*** tmp\_1 **=** t**->**next**;**

**while** **(**tmp\_1 **!=** max**)**

**{**

Stack**\*** tmp\_2 **=** tmp\_1**;**

tmp\_1 **=** tmp\_1**->**next**;**

**delete** tmp\_2**;**

del\_numbs**++;**

**}**

t**->**next **=** max**;**

t **=** b**;** //Удаление вершины

b **=** b**->**next**;**

**delete** t**;**

cout **<<** "Кол-во удаленных элементов: " **<<** del\_numbs **<<** endl**;**

**return** b**;**

**}**

p **=** t**;**

t **=** t**->**next**;**

**}**

t **=** b**;** //Удаление вершины

b **=** b**->**next**;**

**delete** t**;**

cout **<<** "Кол-во удаленных элементов: " **<<** del\_numbs **<<** endl**;**

**return** b**;**

**}**

Stack**\*** mas**(**Stack **\***b**)**

**{**

int n **=** 0**,** **\***a**,** i**,** m**,** min**,** max**,** numb\_of\_elements**=**0**;**

Stack **\***t **=** b**;**

**while** **(**t **!=** **NULL)**

**{**

n**++;**

t **=** t**->**next**;**

**}**

a **=** **new** int**[**n**];**

**for** **(**i **=** 0**;** i **<** n**;** i**++)**

b **=** out\_stack**(**b**,** a **+** i**);**

max **=** a**[**0**];** int pos\_max **=** 0**;**

**for** **(**i **=** 0**;** i **<** n**;** i**++)**

**{**

**if** **(**a**[**i**]** **>** max**)**

**{**

max **=** a**[**i**];**

pos\_max **=** i**;**

**}**

**}**

min **=** a**[**0**];** int pos\_min **=** 0**;**

**for** **(**i **=** 0**;** i **<** n**;** i**++)**

**{**

**if** **(**a**[**i**]** **<** min**)**

**{**

min **=** a**[**i**];**

pos\_min **=** i**;**

**}**

**}**

int pos\_now **=** 0**;**

**if** **(**pos\_min **<** pos\_max**)**

**{**

**for** **(**m **=** i **=** 0**;** i **<** n**;** i**++)**

**{**

pos\_now **=** i**;**

**if** **(**pos\_now **!=** pos\_min**)**

a**[**m**++]** **=** a**[**i**];**

**if** **(**pos\_now **==** pos\_min**)**

**{**

a**[**m**++]** **=** a**[**i**];**

i**++;** pos\_now **=** i**;**

**do**

**{**

**for** **(**int k **=** i**;** k **<** n**;** k**++)**

**{**

a**[**k**]** **=** a**[**k **+** 1**];**

**}**

pos\_max**--;** n**--;** numb\_of\_elements**++;**

**}** **while** **(**pos\_now **!=** pos\_max**);**

a**[**m**++]** **=** a**[**i**];**

**}**

**}**

**}**

**if** **(**pos\_min **>** pos\_max**)**

**{**

**for** **(**m **=** i **=** 0**;** i **<** n**;** i**++)**

**{**

pos\_now **=** i**;**

**if** **(**pos\_now **!=** pos\_max**)**

a**[**m**++]** **=** a**[**i**];**

**if** **(**pos\_now **==** pos\_max**)**

**{**

a**[**m**++]** **=** a**[**i**];**

i**++;** pos\_now **=** i**;**

**do**

**{**

**for** **(**int k **=** i**;** k **<** n**;** k**++)**

**{**

a**[**k**]** **=** a**[**k **+** 1**];**

**}**

pos\_min**--;** n**--;** numb\_of\_elements**++;**

**}** **while** **(**pos\_now **!=** pos\_min**);**

a**[**m**++]** **=** a**[**i**];**

**}**

**}**

**}**

**for** **(**i **=** 0**;** i **<** m**;** i**++)**

b**=** input**(**b**,** a**[**i**]);**

**for** **(**i **=** 0**;** i **<** m**;** i**++)**

b**=** input**(**b**,** a**[**i**]);**

**for** **(**i **=** 0**;** i **<** m**;** i**++)**

b**=** out\_stack**(**b**,** **(**a **+** i**));**

**for** **(**i **=** 0**;** i **<** n**;** i**++)**

b **=** delete\_stack**(**b**);**

**for** **(**i **=** 0**;** i **<** m**;** i**++)**

b **=** input**(**b**,** a**[**i**]);**

cout **<<** endl**;**

cout **<<** "Количество удаленных элементов: " **<<**numb\_of\_elements**;**

**delete[]** a**;**

**return** b**;**

**}**

int main**()**

**{**

setlocale**(**LC\_ALL**,** "rus"**);**

Stack **\***begin**,** **\***t**;**

begin **=** **NULL;**

t**=new** Stack**;**

int a**,** n**,** x**,** out**;**

bool menu **=** **true;**

bool flag **=** **false;**

char sw**;**

cout **<<** "Введите количество элементов в стеке: "**;**

**while** **(!(**cin **>>** n**)** **or** **(**n **<** 1**))**

**{**

cout **<<** "Введено недопустимое значение. Введите значение заново." **<<** endl**;**

cin**.**clear**();**

cin**.**ignore**();**

**}**

cout **<<** endl**;**

cout **<<** "Выбор ввода:" **<<** endl**;**

cout **<<** "1)С клавиатуры" **<<** endl**;**

cout **<<** "2)Рандомом" **<<** endl**;**

**do**

**{**

sw **=** \_getch**();**

**switch** **(**sw**)**

**{**

**case** '1'**:**

**{**

cout **<<** "Ввод элементов стека: " **<<** endl**;**

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

**{**

**while** **(!(**cin **>>**a**))**

**{**

cout **<<** "Введено недопустимое значение. Введите значение заново." **<<** endl**;**

cin**.**clear**();**

cin**.**ignore**();**

**}**

begin **=** input**(**begin**,** a**);**

**}**

flag **=** **true;**

**break;**

**}**

**case** '2'**:**

**{**

srand**(**time**(NULL));** //использует системное время для того, чтобы числа при перезапуске программы отличались

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

**{**

a **=** rand**()** **%** 21 **-** 10**;**

begin **=** input**(**begin**,** a**);**

**}**

cout **<<** endl**;**

flag **=** **true;**

**break;**

**}**

**default:**

**{**

flag **=** **false;**

**}**

**}**

**}** **while** **(**flag **==** **false);**

cout **<<** endl**;**

cout **<<** "Вывод стека: " **<<** endl**;**

**if** **(**begin **==** **NULL)**

cout **<<** "Стек пуст" **<<** endl**;**

**else**

view**(**begin**);**

**while** **(**menu**==** **true)**

**{**

cout **<<** "\t\t\t\t\t\t\t Меню \n" **<<** endl**;**

cout **<<** "1)Ввод элементов стека" **<<** endl**;**

cout **<<** "2)Вывод стека" **<<** endl**;**

cout **<<** "3)Добавление элемента в стек" **<<** endl**;**

cout **<<** "4)Очистка стека" **<<** endl**;**

cout **<<** "5)Получение информации из вершины стека с извлечением" **<<** endl**;**

cout **<<** "6)Выполнение задания используя динамический массив" **<<** endl**;**

cout **<<** "7)Выполнение задания" **<<** endl**;**

cout **<<** "8)Завершить работу программы" **<<** endl**;**

cout **<<** endl**;**

**do**

**{**

sw **=** \_getch**();**

**switch** **(**sw**)**

**{**

**case** '1'**:**

**{**

cout **<<** "Выбор ввода:" **<<** endl**;**

cout **<<** "1)С клавиатуры" **<<** endl**;**

cout **<<** "2)Рандомом" **<<** endl**;**

**do**

**{**

sw **=** \_getch**();**

**switch** **(**sw**)**

**{**

**case** '1'**:**

**{**

cout **<<** "Ввод элементов стека: " **<<** endl**;**

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

**{**

**while** **(!(**cin **>>** a**))**

**{**

cout **<<** "Введено недопустимое значение. Введите значение заново." **<<** endl**;**

cin**.**clear**();**

cin**.**ignore**();**

**}**

begin **=** input**(**begin**,** a**);**

**}**

flag **=** **true;**

**break;**

**}**

**case** '2'**:**

**{**

srand**(**time**(NULL));** //использует системное время для того, чтобы числа при перезапуске программы отличались

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

**{**

a **=** rand**()** **%** 21 **-** 10**;**

begin **=** input**(**begin**,** a**);**

**}**

cout **<<** endl**;**

flag **=** **true;**

**break;**

**}**

**default:**

**{**

flag **=** **false;**

**}**

**}**

**}** **while** **(**flag **==** **false);**

**}**

**case** '2'**:**

**{**

cout **<<** "Вывод стека: " **<<** endl**;**

**if** **(**begin **==** **NULL)**

**{**

cout **<<** "Стек пуст" **<<** endl**;**

**}**

**else**

**{**

view**(**begin**);**

**}**

menu **=** **true;**

flag **=** **true;**

cout **<<** endl**;**

cout **<<** "Выберите новую операцию:" **<<** endl**;**

cout **<<** endl**;**

**break;**

**}**

**case** '3'**:**

**{**

cout **<<** "Добавить элемент в стек: "**;**

**while** **(!(**cin **>>** x**))**

**{**

cout **<<** "Введено недопустимое значение. Введите значение заново." **<<** endl**;**

cin**.**clear**();**

cin**.**ignore**();**

**}**

begin **=** input**(**begin**,** x**);**

menu **=** **true;**

flag **=** **true;**

cout **<<** endl**;**

cout **<<** "Выберите новую операцию:" **<<** endl**;**

cout **<<** endl**;**

**break;**

**}**

**case** '4'**:**

**{**

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

**{**

begin **=** delete\_stack**(**begin**);**

**}**

menu **=** **true;**

flag **=** **true;**

cout **<<** endl**;**

cout **<<** "Выберите новую операцию:" **<<** endl**;**

cout **<<** endl**;**

**break;**

**}**

**case** '5'**:**

**{**

**if** **(**begin **==** **NULL)**

cout **<<** "Стек пуст" **<<** endl**;**

**else**

**{**

begin **=** out\_stack**(**begin**,** **&**out**);**

cout **<<** "Вершина стека: " **<<** out **<<** endl**;**

**}**

menu **=** **true;**

flag **=** **true;**

cout **<<** endl**;**

cout **<<** "Выберите новую операцию:" **<<** endl**;**

cout **<<** endl**;**

**break;**

**}**

**case** '6'**:**

**{**

cout **<<** "Исходный стек:" **<<** endl**;**

view**(**begin**);**

begin **=** mas**(**begin**);**

cout **<<** endl**;**

cout **<<** "Отсортированный стек: " **<<** endl**;**

view**(**begin**);**

flag **=** **true;**

menu **=** **true;**

cout **<<** endl**;**

cout **<<** "Выберите новую операцию:" **<<** endl**;**

cout **<<** endl**;**

**break;**

**}**

**case** '7'**:**

**{**

cout **<<** "Исходный стек:" **<<** endl**;**

view**(**begin**);**

begin **=** Work**(**begin**);**

cout **<<** endl**;**

cout **<<** "Отсортированный стек: " **<<** endl**;**

view**(**begin**);**

flag **=** **true;**

menu **=** **true;**

cout **<<** endl**;**

cout **<<** "Выберите новую операцию:" **<<** endl**;**

cout **<<** endl**;**

**break;**

**}**

**case** '8'**:**

**{**

flag **=** **true;**

menu **=** **false;**

**break;**

**}**

**default:**

**{**

flag **=** **false;**

menu **=** **true;**

**break;**

**}**

**}**

**}** **while** **(**flag **==** **false);**

**}**

**}**

Вывод: создал программу для решения данной задачи.

