Міністерство освіти і науки України  
Черкаський державний технологічний університет

Кафедра програмного забезпечення автоматизованих систем

**Звіт**

Про виконання лабораторної роботи №3

з дисципліни “Алгоритми та структури даних”

|  |  |
| --- | --- |
|  | Виконав:  Студент 2-го курсу  Групи ПЗС-1944  Білозор Д.О.  Перевірив:  Асистент Кафедри ПЗАС Олексюк В.В. |

Черкаси 2019

**Лабораторна робота №7**

**Тема роботи:** Стеки, деки, черги.

**Мета роботи:** Набути навиків роботи зі стеками, деками, чергами.

**Завдання(Варіант 7):**

+додавання елемента в початок дека;

\* Витяг елемента з початку дека;

• додавання елемента в кінець дека;

\* Витяг елемента з кінця дека.**Теоретичні відомості**

На практиці часто використовуються структури даних, в кото -

виняток, включення і доступ майже завжди виробляються

в першому чи останньому вузлах.< :

Стек (stack) - динамічна структура даних, в якій все

включення, виключення і доступ проводяться тільки з одного

кінця за принципом LIFO (від англ. «Last In - First Out» - останнім

прийшов, першим пішов). Стеки широко застосовуються в программи

для: передачі параметрів у функції, трансляції, реалі

рекурсії, реалізації управління динамічною пам'яттю та

ін. Принцип роботи стека легко зрозуміти на прикладі стопки тарілок:

щоб взяти другу зверху, потрібно зняти верхню.

Черга (queue) - динамічна структура даних, в якій

всі включення виробляються на одному кінці, а всі винятки

(і доступ) на іншому за принципом FIFO (від англ. «First In -

First Out» - першим прийшов, першим пішов). Черги знаходять

широке застосування в операційних системах (буфер введення з

клавіатури, буфера вводу-виводу, черги завдань). Часто викорис

ють кільцеві черги, які призначені для зберігання

числа елементів не більше заданого. Прикладом може служити

список раніше відкритих документів або проектів у прикладній

програма.

Дек, він же черга з двома кінцями (deque, double ended queue),

- динамічна структура даних, в якій все включення і ви

ключення (і доступ) робляться на обох кінцях списку. Розрізняти

деки з обмеженим входом і обмеженим виходом. В деці з

обмеженим входом додавання можливо тільки з одного кон

ца, видалення з обох, в деці з обмеженим виходом видалення

можливо тільки з одного кінця, додавання з обох.

Лістинг програми

**Завдання 1**

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

#include <math.h>

#include <windows.h>

#include <deque>

using namespace std;

const int N=5; //размер дека

struct Deque

{

int data[N]; //массив данных

int last; //указатель на конец

};

void Creation(Deque \*D) //создание дека

{ D->last=0; }

bool Full(Deque \*D) //проверка дека на пустоту

{

if (D->last==0) return true;

else return false;

}

void AddL(Deque \*D) //добавление элемента в начало

{

if (D->last==N)

{ cout<<"\nДек заполнен\n\n"; return; }

int value;

cout<<"\nЗначение > "; cin>>value;

for (int i=D->last; i>0; i--)

D->data[i]=D->data[i-1];

D->data[0]=value;

D->last++;

cout<<endl<<"Элемент добавлен\n\n";

}

void AddR(Deque \*D) //добавление элемента в конец

{

if (D->last==N)

{ cout<<"\nДек заполнен\n\n"; return; }

int value;

cout<<"\nЗначение > "; cin>>value;

D->data[D->last++]=value;

cout<<endl<<"Элемент добавлен\n\n";

}

void DeleteL(Deque \*D) //удаление первого элемента

{

for (int i=0; i<D->last; i++) //смещение элементов

D->data[i]=D->data[i+1]; D->last--;

}

void DeleteR(Deque \*D) //удаление последнего элемента

{ D->last--; }

int OutputL(Deque \*D) //вывод первого элемента

{ return D->data[0]; }

int OutputR(Deque \*D) //вывод последнего элемента

{ return D->data[D->last-1]; }

int Size(Deque \*D) //размер дека

{ return D->last; }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main() //главная функция

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Deque D;

Creation(&D);

char number;

do

{

cout<<"1. Добавить элемент в начало дека"<<endl;

cout<<"2. Добавить элемент в конец дека"<<endl;

cout<<"3. Удалить первый элемент дека"<<endl;

cout<<"4. Удалить последний элемент дека"<<endl;

cout<<"5. Вывести первый элемент дека"<<endl;

cout<<"6. Вывести последний элемент дека"<<endl;

cout<<"7. Узнать размер дека"<<endl;

cout<<"0. Выйти из программы\n\n";

cout<<"Номер команды > "; cin>>number;

switch (number)

{

case '1': AddL(&D);

break;

//-----------------------------------------------

case '2': AddR(&D);

break;

//-----------------------------------------------

case '3':

if (Full(&D)) cout<<endl<<"Дек пуст\n\n";

else

{

DeleteL(&D);

cout<<endl<<"Элемент удален из дека\n\n";

} break;

//-----------------------------------------------

case '4':

if (Full(&D)) cout<<endl<<"Дек пуст\n\n";

else

{

DeleteR(&D);

cout<<endl<<"Элемент удален\n\n";

} break;

//-----------------------------------------------

case '5':

if (Full(&D)) cout<<endl<<"Дек пуст\n\n";

else cout<<"\nПервый элемент: "<<OutputL(&D)<<"\n\n";

break;

//-----------------------------------------------

case '6':

if (Full(&D)) cout<<endl<<"Дек пуст\n\n";

else cout<<"\nПоследний элемент: "<<OutputR(&D)<<"\n\n";

break;

//-----------------------------------------------

case '7':

if (Full(&D)) cout<<endl<<"Дек пуст\n\n";

else cout<<"\nРазмер дека: "<<Size(&D)<<"\n\n";

break;

//-----------------------------------------------

case '0': break;

default: cout<<endl<<"Команда не определена\n\n";

break;

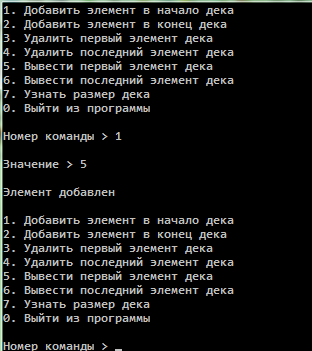
}

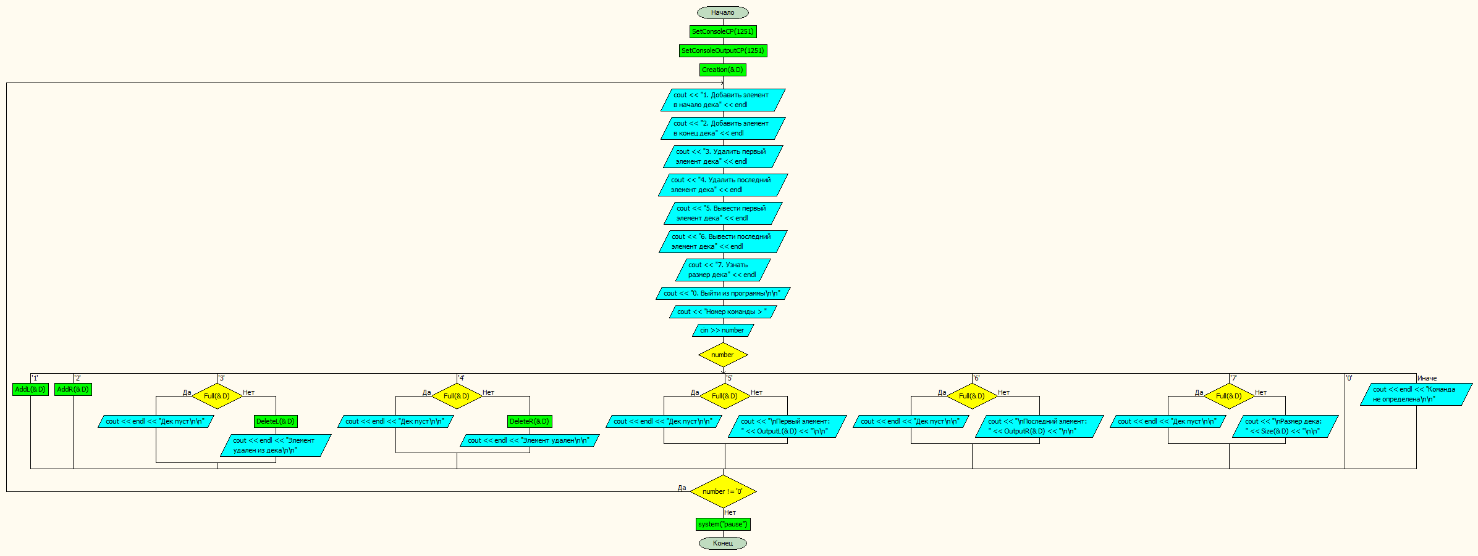
} while(number!='0');

system("pause");

}**Результат роботи**

**Завдання 1**



**Завдання 2. Блок-схема:**

Висновок

На даній лабораторній роботі я набув навиків роботи зі стеками, деками, чергами. Я набув навичок створення і збереження стеків, деків, черг та їх оброблювати. Програма містить меню вибору.