**Лабораторная работа №9. Полустатические структуры данных: очереди**

**Вариант 10**

Условие:

Разработать функцию, которая в очереди переставляет в обратном порядке все элементы между первыми последним  вхождением элемента  E, если E входит в L не менее двух раз.

Код(главный файл):

#include "stdafx.h"

#include "list.h"

#include <iostream>

using namespace std;

void main()

{ setlocale(LC\_ALL, "Rus");

Queue Q;

Creation(&Q);

int number;

do

{ cout << "1. Добавить элемент" << endl;

cout << "2. Удалить элемент" << endl;

cout << "3. Вывести верхний элемент" << endl;

cout << "4. Узнать размер очереди" << endl;

cout << "5. Выввести очередь" << endl;

cout << "6. Реверс очереди" << endl;

cout << "0. Выйти\n\n";

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

switch (number)

{ case 1: Add(&Q); break;

case 2: if (Full(&Q)) cout << endl << "Очередь пуста\n\n";

else

{ Delete(&Q);

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

} break;

case 3: if (Full(&Q)) cout << endl << "Очередь пуста\n\n";

else cout << "\nНачальный элемент: " << Top(&Q) << "\n\n";

break;

case 4: if (Full(&Q)) cout << endl << "Очередь пуста\n\n";

else cout << "\nРазмер очереди: " << Size(&Q) << "\n\n";

break;

case 5: vivod(&Q); break;

case 6: revers(&Q); break;

case 0: break;

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

break;

}

} while (number != 0);

}

Код(isx.cpp):

#include "stdafx.h"

#include "list.h"

#include <iostream>

using namespace std;

void Creation(Queue \*Q) //Создание очереди Q

{ Q->last = 0;

}

void revers(Queue \*Q)

{

int rev[N], i=0,j=0;

for(j=Q->last-1;j>=0;j--,i++)

{

rev[i]=Q->data[j];

cout<<rev[i]<<" ";

}

cout<<endl;

}

bool Full(Queue \*Q) //Проверка очереди на пустоту

{ if (Q->last == 0) return true;

else return false;

}

void vivod(Queue \*Q)

{

int i=0;

for(i;i<Q->last;i++)

cout<<Q->data[i]<<" ";

cout<<endl;

}

void Add(Queue \*Q) //Добавление элемента

{ if (Q->last == N)

{ cout << "\nОчередь заполнена\n\n"; return; }

int value;

cout << "\nЗначение > ";

cin >> value;

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

cout << endl << "Элемент добавлен в очередь\n\n";

}

void Delete(Queue \*Q) //Удаление элемента

{ for (int i = 0; i<Q->last && i < N; i++) //смещение элементов

Q->data[i] = Q->data[i + 1];

Q->last--;

}

int Top(Queue \*Q) //Вывод начального элемента

{ return Q->data[0];

}

int Size(Queue \*Q) //Размер очереди

{ return Q->last;

}

Код(list.h):

#include "stdafx.h" //размер очереди

const int N = 4;

struct Queue

{ int data[N];//массив данных

int last; //указатель на начало

};

void Creation(Queue \*Q);

bool Full(Queue \*Q);

void Add(Queue \*Q);

void Delete(Queue \*Q) ;

int Top(Queue \*Q);

int Size(Queue \*Q);

void vivod(Queue \*Q);

void revers(Queue \*Q);

