Белорусский государственный университет информатики и радиоэлектроники

Факультет компьютерных технологий

ОТЧЕТ

по лабораторной работе №4

на тему: **Динамическая структура ОЧЕРЕДЬ**

|  |  |
| --- | --- |
| Выполнил  студент  группы №980161  Алейчик И.Д. | Проверил  доцент  Казакевич Ю.Д. |

Минск 2020

**Лабораторная работа №4**

**Вариант 1**

**Тема:**

Динамическая структура ОЧЕРЕДЬ

**Цель работы:**

Изучить возможности работы со списками, организован-ными в виде очереди.

**Краткие теоретические сведения:**

Очередь–линейный список, в котором извлечение данных происходит из начала, а добавление–в конец, т. е.этоструктура, организованнаяпо принципу FIFO(First In, First Out) –первым вошел, первымвыйдет. При работе с очередью используют два указателя–на первый элемент (начало–begin) и на последний (конец–end). Очереди организуются в виде односвязных или двухсвязных списков, в зависимости от количества связей (указателей) в адресной части элемента структуры.

Двухсвязным(двунаправленным) является список, в адресную частько-торого кроме указателя на следующий элемент включен и указатель на преды-дущий. Зададимструктуру, в которой адресная часть состоит из указателей на предыдущий (prev) и следующий (next) элементы:struct Spis2 {int info;Spis2 \*prev, \*next;}\*begin, \*end;Формирование двунаправленного списка проводится в два этапа: форми-рованиепервого элемента и добавление нового, причем добавление может вы-полняться как в начало(begin), так и в конец (end) списка.

**Задание:**

Написать программу по созданию, добавлению(в начало, в конец), про-смотру(с начала, с конца) и решению приведенной в подразделе 3.3 задачи для двунаправленных линейных списков.

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

1. Разделить созданный список на два: в первом – положительные числа, вовтором – отрицательные.

**Ход выполнения работы:**

struct Mlist //Структура, являющаяся звеном списка

{

int info; //Значение x будет передаваться в список

Mlist\* Next, \* Prev; //Указатели на адреса следующего и предыдущего элементов списка

};

void createMaxMinListAndView(Mlist\*b, Mlist\*e)

{

Mlist\* temp = b;

while (temp != NULL)

{

if (temp->info > 0) {

cout << temp->info << " ";

}

temp = temp->Next;

}

cout << "\n";

temp = b;

while (temp != NULL)

{

if (temp->info < 0) {

cout << temp->info << " ";

}

temp = temp->Next;

}

cout << "\n\n";

temp = e;

while (temp != NULL)

{

if (temp->info > 0) {

cout << temp->info << " ";

}

temp = temp->Prev;

}

cout << "\n";

temp = e;

while (temp != NULL)

{

if (temp->info < 0) {

cout << temp->info << " ";

}

temp = temp->Prev;

}

cout << "\n";

}

void Add(int in, Mlist\*\*b, Mlist\*\*e)

{

Mlist\* temp = new Mlist;

temp->Next = NULL;

temp->info = in;

if (\*b != NULL) // если не пустой, добовлем

{

temp->Prev = \*e;

(\*e)->Next = temp;

\*e = temp;

}

else //Если список пустой создаем как новый элемент

{

temp->Prev = NULL;

\*b = \*e = temp;

}

}

void ShowBack(Mlist\*b, Mlist\*e ){

////выводим псиок с конца

Mlist\*temp = e;

while (temp != NULL)

{

cout << temp->info << " ";

temp = temp->Prev;

}

cout << "\n";

////выводим псиок с начала

temp = b;

while (temp != NULL)

{

cout << temp->info << " ";

temp = temp->Next;

}

cout << "\n";

}

void multiList() {

Mlist\* beg = NULL;

Mlist\* end = NULL;

srand(time(0));

for (int i = 0; i < 15; i++)

{

Add((-(rand() % 100) + (rand() % 100)),&beg,&end);

}

ShowBack(beg,end);

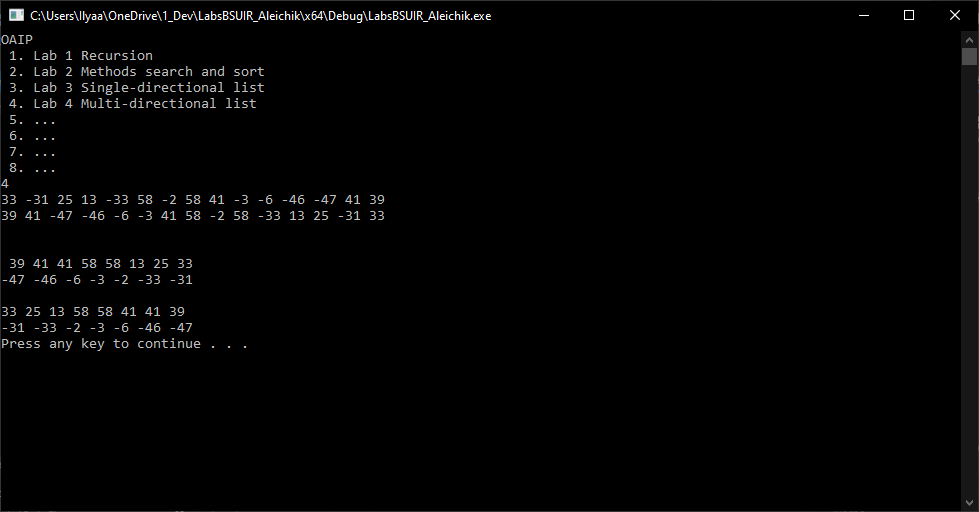
cout << "\n\n ";

createMaxMinListAndView(beg,end);

system("pause");

}

**Результат работы программы:**



**Вывод:**

Изучены возможности работы со списками, организован-ными в виде очереди.