

#pragma once

template<typename type>

class Item

{

public:

type Info;

Item<type>\* Next;

Item(type info, Item\* next) :Info(info), Next(next) {}

};

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

#pragma once

#include "Item.h"

template<typename type>

class Queue

{

public:

Item<type> \* First;

Item<type> \* Last;

Queue() { First = Last = 0; }

void Add(type value) //Додавання нового елемента

{

Item<type> \* current = new Item<type>(value,0);

if (First==0)

{

First = current;

Last = current;

}

else

{

Last->Next = current;

Last = current;

}

}

//type Dequeue();

bool Remove(type& value) //Вилучення елемента (елемент повертаємо через параметр посилання value)

{

if (First==0)

{

return false;

}

else

{

Item<type>\* curerent = First;

value = First->Info;

First = First->Next;

delete curerent;

if (First==0)

{

Last = 0;

}

return true;

}

}

~Queue()

{

type value;

while (Remove(value))

{}

}

};

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

// ConsoleApplication57.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include"Queue.h"

#include<iostream>

using namespace std;

int main()

{

//З клавіатури вводяться числа до тих пір, поки не буде введено 0. Вивести більші за останній у порядку їх введення.

Queue<double>\* queue = new Queue<double>();

double number;

double last;

do

{

printf("number=");

cin >> number;

if (number!=0)

{

queue->Add(number); //Якщо не нуль, то додаємо до черги

last = number; //У last збережемо останній ненульовий елемент

}

} while (number!=0);

printf("-----------------------\n");

while (queue->Remove(number))

{

if (number>last)

{

printf("%f\n",number);

}

}

delete queue;

system("pause");

return 0;

}