Lstack.h and Lstack.cpp is taken in previous project.

**QuStack.h file:**

#pragma once

#include"LStack.h"

class QuStack {

LStack S1, S2;

public:

void Enqueue(int);

void Dequeue();

int front();

friend ostream& operator<<(ostream&, QuStack&);

};

**QuStack.cpp file:**

#include "QuStack.h"

void QuStack::Enqueue(int item) { S1.push(item); }

void QuStack::Dequeue() {

int temp;

while (!S1.isEmpty()) {

temp = S1.top();

S2.push(temp);

S1.pop();

}

S2.pop();

while (S2.isEmpty()) {

temp = S2.top();

S1.push(temp);

S2.pop();

}

}

int QuStack::front() {

int front, temp;

while (!S1.isEmpty()) {

front = S1.top();

S2.push(front);

S1.pop();

}

while (!S2.isEmpty()) {

temp = S2.top();

S1.push(temp);

S2.pop();

}

return front;

}

ostream& operator<<(ostream& out, QuStack& Q) {

out << "Queue from back to front: "<< Q.S1;

return out;

}

**Main.cpp file**:

#include<iostream>

#include"QuStack.h"

using namespace std;

int main() {

QuStack AL;

for (int i = 0; i < 6; i++) {

AL.Enqueue(i);

cout << AL;

cout << "Front: " << AL.front() << endl;

}

for (int i = 0; i < 3; i++) {

AL.Dequeue();

cout << AL;

cout << "Front: " << AL.front() << endl;

}

system("pause");

return 0;

}

