#include <iostream>

#include <vector>

using namespace std;

template<typename T>

class My\_stk{

vector<T> vec\_stk;

public:

void Push(T data){

vec\_stk.push\_back(data);

}

void Pop(){

if(isEmpty())

cout<<" stack is Empty"<<endl;

else

vec\_stk.pop\_back();

}

T top() const{

if(isEmpty())

{cout<<"stack is Empty"<<endl; return static\_cast<T>(0);}

else{

return vec\_stk.at(vec\_stk.size()-1);

}

}

bool isEmpty() const{

if(vec\_stk.size()>0) return false;

else return true;

}

void Print\_stk(){

if(isEmpty()){

cout<<" stack is Empty"<<endl;

return;

}

for(int i=vec\_stk.size();i>0;--i){

cout<<vec\_stk.top() <<" ";

vec\_stk.Pop();

}

cout<<endl;

}

int getSize(){

return vec\_stk.size();

}

};

template<typename Q\_t>

class Queue{

My\_stk<Q\_t> stk1;

My\_stk<Q\_t> stk2;

public:

void dequeue(){//remove element from queue

if(stk2.getSize()<=0){

if(stk1.getSize()>0){

for(int i=stk1.getSize(); i>0;--i)

{

stk2.Push(stk1.top());

stk1.Pop();

}

}

}

stk2.Pop();

}

void enqueue(Q\_t data){

stk1.Push(data);

}

void PrintQueue(){

if(stk1.getSize()>0){

for(int i=stk1.getSize(); i>0;--i)

{

stk2.Push(stk1.top());

stk1.Pop();

}

}

for(int i=stk2.getSize(); i>0;--i){

cout<< stk2.top()<<" ";

stk2.Pop();

}

}

};

int main() {

// your code goes here

/\* My\_stk<int> mstk;

cout<<mstk.top()<<endl;

mstk.Print\_stk();

mstk.Push(100);

mstk.Push(200);

mstk.Push(300);

mstk.Push(400);

mstk.Print\_stk();

mstk.Pop();

mstk.Pop();

mstk.Print\_stk();

cout<<mstk.top()<<endl;\*/

Queue<int> q1;

q1.enqueue(10);

q1.enqueue(20);

q1.enqueue(30);

q1.dequeue();

q1.dequeue();

q1.enqueue(100);

q1.enqueue(200);

q1.dequeue();

q1.dequeue();

q1.PrintQueue();

return 0;

}