## Homework 2 (With Java) --------by 2019300003090 杜沛然

\*\*it’s recommended to paste this page into a markdown editor\*\*

\*\*all codes have been compiled and run , no error reported\*\*

all is published under GPL v3

view source code at github is quite better

[HomePage](https://github.com/dupeiran001/homework)

if you are using a POSIX system, try the following command

```shell

git clone https://github.com/dupeiran001/homework

cd homework

```

every examples can be compiled by

```shell

javac <target.java>

```

and run by

```shell

java <target>

```

in it’s own folder

contect by 2083856903@qq.com if there is any questioin

ofcource, source code are also pasted below

``` java

//question1

public class p2q1

{

public static void main(String[] args)

{

int[] result = {2 , 1 , 3 , 4 , 5 , 6};

int[] input = {1 , 2 , 3 , 4 , 5 , 6};

for(int tmp = 0 ; tmp != input.length ; tmp++ )

System.out.print(input[tmp] + " ");

System.out.println();

for(int tmp = 0 ; tmp != result.length ; tmp++ )

System.out.print(result[tmp] + " ");

System.out.println();

stack st = new stack();

st.push(input[0]);

int tmp;

int rtmp = 0;

for(tmp = 0 ; rtmp < result.length-1 && tmp < input.length-1 ; )

{

if(st.top() == result[rtmp])

{

st.pop();

rtmp++;

}

else

{

tmp++;

st.push(input[tmp]);

}

}

if(st.empty())

System.out.println("true");

else

System.out.println("false");

}

}

class stack

{

private class node

{

private int data;

private node next;

}

private node head;

public stack()

{

this.head = null;

}

public boolean push(int i)

{

if(head == null)

{

this.head = new node();

this.head.data = i;

this.head.next = null;

}

else

{

node tmp = new node();

tmp.data = i;

tmp.next = this.head;

this.head = tmp;

}

return true;

}

public boolean empty()

{

return this.head == null;

}

public int top()

{

if(this.empty())

return 0;

return this.head.data;

}

public boolean pop()

{

if(this.empty())

return false;

this.head = this.head.next;

return true;

}

}

```

``` java

//question2

public class p2q2

{

public static void main(String[] args)

{

char[] st = {'a','b','c','d','e','@','e','d','c','b','a'};

seqstack a = new seqstack();

int tmp;

for(tmp = 0 ; st[tmp] != '@' ; tmp++)

a.push(st[tmp]);

tmp++;

for( ; st[tmp] == a.top() && tmp < st.length-1 ; tmp++)

a.pop();

if(a.top()==st[tmp])

{

a.pop();

if(a.empty())

System.out.println("true");

}

else

System.out.println(tmp);

}

}

class seqstack

{

private int datalength;

private int stringlength;

private char[] data;

public seqstack()

{

this.datalength = 0;

this.stringlength = 10;

this.data = new char[this.stringlength];

}

private void doublecapsity()

{

char[] tmp = new char[2\*this.stringlength];

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

tmp[i] = this.data[i];

this.data = tmp;

}

public void push(char i)

{

if(this.full())

this.doublecapsity();

data[this.datalength] = i;

this.datalength ++;

}

public char top()

{

if(this.empty())

return 0;

return data[datalength-1];

}

public boolean pop()

{

if(this.empty())

return false;

data[datalength-1] = 0;

datalength--;

return true;

}

public boolean empty()

{

return datalength == 0;

}

private boolean full()

{

return datalength == stringlength;

}

}

```

``` java

//question3

public class p2q3

{

public static void main(String[] args)

{

int[] source = {1 , 2 , 3 , 4 , 5 , 6};

for(int i = 0; i != source.length ; i++ )

System.out.println(source[i]);

stack a = new stack();

for(int tmp = 0; tmp != source.length ; tmp++)

a.push(source[tmp]);

for(int b = 0 ; !a.empty() ; b++ )

{

source[b] = a.top();

a.pop();

}

for(int i = 0; i != source.length ; i++ )

System.out.println(source[i]);

}

}

class stack

{

private class node

{

private int data;

private node next;

}

private node head;

public stack()

{

this.head = null;

}

public boolean push(int i)

{

if(head == null)

{

this.head = new node();

this.head.data = i;

this.head.next = null;

}

else

{

node tmp = new node();

tmp.data = i;

tmp.next = this.head;

this.head = tmp;

}

return true;

}

public boolean empty()

{

return this.head == null;

}

public int top()

{

if(this.empty())

return 0;

return this.head.data;

}

public boolean pop()

{

if(this.empty())

return false;

this.head = this.head.next;

return true;

}

}

```

``` java

//question4

public class p2q4

{

public static void main(String[] args)

{

query ood = new query();

query even = new query();

int[] a = {1 , 2 , 3 , 4 , 5 , 6 , 7 , 8};

for(int b : a)

System.out.println(b);

for(int tmp = 0; tmp != a.length ; tmp++)

{

if(a[tmp] % 2 == 0)

even.push(a[tmp]);

else

ood.push(a[tmp]);

}

int tmp;

for(tmp = 0; !even.empty() ; tmp++)

{

a[tmp] = even.top();

even.pop();

}

for( ; !ood.empty() ; tmp++)

{

a[tmp] = ood.top();

ood.pop();

}

System.out.println();

for(int b : a)

System.out.println(b);

}

}

class query

{

private class node

{

private node next;

private int data;

}

private node head;

private node tail;

public query()

{

this.head = null;

this.tail = null;

}

public boolean empty()

{

return this.head == null ;

}

public boolean push(int i)

{

if(this.tail == null)

{

this.tail = new node();

this.tail.data = i;

this.tail.next = null;

this.head = this.tail;

return true;

}

else

{

node tmp = new node();

tmp.data = i;

tmp.next = null;

this.tail.next = tmp;

this.tail = tmp;

return true;

}

}

public int top()

{

if(this.empty())

return 0;

return this.head.data;

}

public boolean pop()

{

if(this.empty())

return false;

else

{

this.head = this.head.next;

return true;

}

}

}

```

``` java

//question5

public class p2q5

{

public static void main(String[] args)

{

cycleque a = new cycleque(1);

a.push(2);

a.push(3);

a.push(4);

a.push(5);

a.push(6);

a.push(7);

a.push(8);

a.push(9);

a.push(10);

System.out.println(a.top());

}

}

class cycleque

{

private int MaxSize;

private boolean tag;

private int front;

private int rear;

private int[] data;

public cycleque(int i)

{

this.MaxSize = 10;

tag = false;

front = 0;

rear = 0;

data = new int[MaxSize];

data[0] = i;

}

public boolean empty()

{

return front == rear && !tag;

}

public boolean full()

{

return (front+1) % MaxSize == rear && tag;

}

private void doublecapsity()

{

int[] a = new int[2\*MaxSize];

int tmp;

for(tmp = 0; front != rear ; tmp++ , rear=(rear++)%MaxSize)

a[tmp] = data[rear];

this.rear = 0;

this.front = tmp-1;

MaxSize \*= 2;

this.data = a;

this.tag = false;

}

public boolean push(int i)

{

if(this.full())

this.doublecapsity();

this.front = (this.front+1) % MaxSize ;

data[this.front] = i;

if(this.rear == this.front)

tag = true;

return true;

}

public boolean pop()

{

if(this.empty())

return false;

this.data[this.rear] = 0;

this.rear = (this.rear+1) % MaxSize;

if(this.rear == this.front)

tag = false;

return true;

}

public int top()

{

if(this.empty())

return 0;

return this.data[rear];

}

}

```

``` java

//question6

public class p2q6

{

public static void main(String[] args)

{

query q = new query();

q.push(1);

q.push(2);

q.push(3);

System.out.println(q.top());

q.pop();

System.out.println(q.top());

}

}

class query

{

private stack stack1;

private stack stack2;

public query()

{

stack1 = new stack();

stack2 = new stack();

}

public boolean push(int i)

{

return stack1.push(i);

}

public boolean empty()

{

if(stack1.empty()&&stack2.empty())

return true;

return false;

}

public void reverse()

{

if(stack2.empty())

{

while(!stack1.empty())

{

stack2.push(stack1.top());

stack1.pop();

}

}

}

public int top()

{

if(stack2.empty())

this.reverse();

return this.stack2.top();

}

public boolean pop()

{

if(this.empty())

return false;

else if(stack2.empty())

this.reverse();

this.stack2.pop();

return true;

}

}

class stack

{

private class node

{

private int data;

private node next;

}

private node head;

public stack()

{

this.head = null;

}

public boolean push(int i)

{

if(head == null)

{

this.head = new node();

this.head.data = i;

this.head.next = null;

}

else

{

node tmp = new node();

tmp.data = i;

tmp.next = this.head;

this.head = tmp;

}

return true;

}

public boolean empty()

{

return this.head == null;

}

public int top()

{

if(this.empty())

return 0;

return this.head.data;

}

public boolean pop()

{

if(this.empty())

return false;

this.head = this.head.next;

return true;

}

}

```

``` java

//question7

public class p2q7

{

public static void main(String[] args)

{

stack s = new stack();

s.push(1);

s.push(2);

s.push(3);

s.push(4);

s.pop();

System.out.println(s.top());

}

}

class stack

{

private query full;

private query empty;

private int count;

public stack()

{

full = new query();

empty = new query();

count = 0;

}

public boolean empty()

{

return this.count == 0;

}

public boolean push(int i)

{

count++;

full.push(i);

return true;

}

public boolean pop()

{

int num = count;

if(this.empty())

return false;

while(num!=1)

{

empty.push(full.top());

full.pop();

num--;

}

full.pop();

query tmp = full;

full = empty;

empty = tmp;

count--;

return true;

}

public int top()

{

int tmp = 0;

if(this.empty())

return 0;

while(!full.empty())

{

empty.push(full.top());

tmp = full.top();

full.pop();

}

query q = full;

full = empty;

empty = q;

return tmp;

}

}

class query

{

private class node

{

private node next;

private int data;

}

private node head;

private node tail;

public query()

{

this.head = null;

this.tail = null;

}

public boolean empty()

{

return this.head == null ;

}

public boolean push(int i)

{

if(this.tail == null)

{

this.tail = new node();

this.tail.data = i;

this.tail.next = null;

this.head = this.tail;

return true;

}

else

{

node tmp = new node();

tmp.data = i;

tmp.next = null;

this.tail.next = tmp;

this.tail = tmp;

return true;

}

}

public int top()

{

if(this.empty())

return 0;

return this.head.data;

}

public boolean pop()

{

if(this.empty())

return false;

else

{

this.head = this.head.next;

return true;

}

}

}

```