4. Design a Java interface for ADT Stack. Implement this interface using array. Provide necessary exception handling in both the implementations.

**PROGRAM**

import java.io.\*;

interface stackoperation

{

public void push(int i);

public void pop();

}

class Astack implements stackoperation

{

int stack[]=new int[5];

int top=-1;

int i;

public void push(int item)

{

if(top>=4)

{

System.out.println("overflow");

}

else

{

top=top+1;

stack[top]=item;

System.out.println("item pushed"+stack[top]);

}

}

public void pop()

{

if(top<0)

System.out.println("underflow");

else

{

System.out.println("item popped"+stack[top]);

top=top-1;

}

}

public void display()

{

if(top<0)

System.out.println("No Element in stack");

else

{

for(i=0;i<=top;i++)

System.out.println("element:"+stack[i]);

}

}

}

class teststack

{

public static void main(String args[])throws IOException

{

int ch,c;

int i;

Astack s=new Astack();

DataInputStream in=new DataInputStream(System.in);

do

{

try

{

System.out.println("ARRAY STACK");

System.out.println("1.push 2.pop 3.display 4.exit");

System.out.println("enter ur choice:");

ch=Integer.parseInt(in.readLine());

switch(ch)

{

case 1:

System.out.println("enter the value to push:");

i=Integer.parseInt(in.readLine());

s.push(i);

break;

case 2:

s.pop();

break;

case 3:

System.out.println("the elements are:");

s.display();

break;

case 4:

break;

}

}

catch(IOException e)

{

System.out.println("io error");

}

System.out.println("Do u want to continue 0 to quit and 1 to continue ");

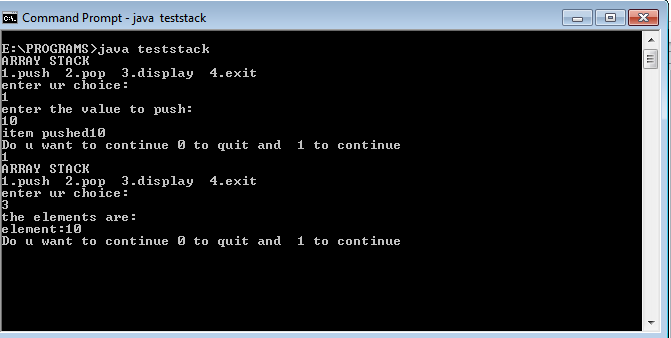
c=Integer.parseInt(in.readLine());

}while(c==1);

}

}

**OUTPUT**

****

