import java.util.\*;

import java.io.\*;

interface Stackinterface

{

void push(int n1) throws Exception;

void pop() throws Exception;

void peek() throws Exception;

boolean isempty();

boolean isfull();

void display() throws Exception;

}

class StackemptyException extends Exception

{

public StackemptyException(){}

public String toString()

{

return ("STACK UNDERFLOW");

}

}

class StackFullException extends Exception

{

int n;

public StackFullException(int n1)

{

n=n1;

}

public String toString()

{

return ("Stack Overflow! Cannot push "+n);

}

}

class arrayStack implements Stackinterface

{

int[] a;

int size,capacity;

public arrayStack(int n1)

{

a=new int[n1];

size=0;

capacity=n1;

}

public void push(int n1) throws Exception

{

if(isfull())

throw new StackFullException(n1);

else

a[size++]=n1;

}

public void pop() throws Exception

{

if(isempty())

throw new StackemptyException();

else

size--;

}

public void peek() throws Exception

{

if(isempty())

throw new StackemptyException();

else

System.out.println("Top of stack : "+a[size-1]);

}

public boolean isempty()

{

if(size==0)

return true;

else

return false;

}

public boolean isfull()

{

if(size==capacity)

return true;

else

return false;

}

public void display() throws Exception

{

if(isempty())

throw new StackemptyException();

else

{

System.out.print("[");

for(int i=size-1; i>=0; i--)

{

System.out.print(a[i]);

if(i!=0)

System.out.print(", ");

}

System.out.print("]");

}

System.out.print("\n");

}

}

class node

{

private int element;

node next;

public node(int e)

{

element=e;

next=null;

}

public void display()

{

System.out.println(element);

}

}

class listStack implements Stackinterface

{

private node first;

public listStack()

{

first=null;

}

public boolean isempty()

{

if(first==null)

return true;

else

return false;

}

public boolean isfull()

{

System.out.println("there is no limit on size!");

return false;

}

public void push(int e)

{

node temp=new node(e);

temp.next=first;

first=temp;

}

public void pop() throws Exception

{

if(!isempty())

{

node temp=first;

first=first.next;

}

else

throw new StackemptyException();

}

public void peek() throws Exception

{

if(!isempty())

{

node temp=first;

System.out.println("Top of Stack:");

temp.display();

}

else

throw new StackemptyException();

}

public void display()

{

System.out.println("Stack:");

node temp=first;

while(temp!=null)

{

temp.display();

temp=temp.next;

}

}

}

public class Stacka

{

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

{

Stackinterface s;

int n,ch,choice;

boolean b;

Scanner in=new Scanner(System.in);

do

{

System.out.println("\nStack Implementation!\n1.Arrays\n2.Linkedlist\n3.Exit\nEnter your choice");

choice=in.nextInt();

if(choice==1)

{

System.out.println("\nEnter the number o felements in the stack\n");

n=in.nextInt();

s=new arrayStack(n);

}

else if(choice==2)

{

s=new listStack();

}

else if(choice==3)

break;

else

continue;

if(choice==1||choice==2)

{

do

{

System.out.println("Stack operations\n1.Push\n2.Pop\n3.peek\n4.isempty?\n5.isfull?\n6.display\n7.Exit\nEnter your choice:");

ch=in.nextInt();

switch(ch)

{

case 1:try

{

System.out.println("Enter the number to be pushed : ");

n=in.nextInt();

s.push(n);

s.display();

}

catch(StackFullException sf)

{

System.out.println(sf);

}

break;

case 2:try

{

s.pop();

s.display();

}

catch(StackemptyException se)

{

System.out.println(se);

}

break;

case 3:try

{

s.peek();

}

catch(StackemptyException se)

{

System.out.println(se);

}

break;

case 4:b=s.isempty();

if(b)

System.out.println("Stack empty!");

else

System.out.println("Stack not empty!");

break;

case 5:b=s.isfull();

if(b)

System.out.println("Stack Full!");

else

System.out.println("Stack not Full!");

break;

case 6:try

{

s.display();

}

catch(StackemptyException se)

{

System.out.println(se);

}

break;

case 7:break;

}

}while(ch!=7);

}

}while(choice!=3);

}

}

SAMPLE INPUT AND OUTPUT:

Stack Implementation!

1.Arrays

2.Linkedlist

3.Exit

Enter your choice

1

Enter the number o fel

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number to be

1

[1]

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number to be

2

[2, 1]

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number to be

5

[5, 2, 1]

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number to be

6

[6, 5, 2, 1]

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number to be

4

Stack Overflow! Cannot

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

3

Top of stack : 6

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

4

Stack not empty!

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

5

Stack Full!

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

6

[6, 5, 2, 1]

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

[5, 2, 1]

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

[2, 1]

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

[1]

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

STACK UNDERFLOW

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

STACK UNDERFLOW

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

7

Stack Implementati

1.Arrays

2.Linkedlist

3.Exit

Enter your choice

2

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number t

4

Stack:

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number t

5

Stack:

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number t

7

Stack:

7

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number t

3

Stack:

3

7

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number t

6

Stack:

6

3

7

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

1

Enter the number t

8

Stack:

8

6

3

7

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

3

Top of Stack:

8

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

4

Stack not empty!

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

5

there is no limit

Stack not Full!

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

6

Stack:

8

6

3

7

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

Stack:

6

3

7

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

Stack:

3

7

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

Stack:

7

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

Stack:

5

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

Stack:

4

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

Stack:

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:

2

STACK UNDERFLOW

Stack operations

1.Push

2.Pop

3.peek

4.isempty?

5.isfull?

6.display

7.Exit

Enter your choice:7