**I Practice Stack**

**1.Question**

**Stack 1 : Array Implementation : Push and Pop**

Consider implementing a fixed size stack of maximum size 5 using an array.

Create a structure

struct stack {

int a[5];

int top;

};

**Note :**  
The array a holds the contents of the stack and the integer top stores the index of the element at the top of the stack.

Write a program to implement push and pop operation on stack and to display the contents of the stack.

In the initstack function intialize the value of top to -1.

Print the message “Stack is full” in the push function when an attempt is made to push a data into a full stack.

Print the message “Stack is empty” in the pop function and return the value -1000 when an attempt is made to pop data from an empty stack.

Refer function specifications for further details.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

**Note :**  
 The statement “The contents of the stack are” is in the main function.  
  In the display function, if the stack is empty, print “ **{}**”.

[All text in bold corresponds to input and the rest corresponds to output]

**Sample Input and Output:**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**3**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**3**

The contents of the stack are 3

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**5**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

The popped element is 5

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

The popped element is 3

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

Stack is empty

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**1**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**2**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**3**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**4**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**5**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**6**

Stack is full

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**3**

The contents of the stack are 1 2 3 4 5

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

The popped element is 5

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**3**

The contents of the stack are 1 2 3 4

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

The popped element is 4

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**3**

The contents of the stack are 1 2 3

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**5**

**Program:**

import java.io.\*;  
import java.util.Scanner;  
class Stack1{  
    int top=-1;  
    int[] arr=new int[5];  
     
    public void push1(int data){  
        if(top==arr.length-1){  
            System.out.println("Stack is Full");  
        }else{  
            top++;  
            arr[top]=data;  
        }  
    }  
    public void pop1(){  
        if(top==-1){  
            System.out.println("Stack is empty");  
        }else{  
            System.out.println("The popped element is "+arr[top]);  
            int temp=top;  
            top--;  
            temp=0;  
        }  
    }  
    public void display1(){  
        if(top==-1){  
            System.out.println("Stack is Empty");  
        }else{  
            for(int i=0;i<=top;i++){  
                 
                System.out.println(arr[i]);  
            }  
        }  
    }  
}  
  
  
  
public class Main{  
    public static void main(String args[]){  
        Scanner sc=new Scanner(System.in);  
         int ch,data;  
        Stack1 s1=new Stack1();  
        do{  
         
       System.out.print("Choice 1 : Push\n"+  
                        "Choice 2 : Pop\n"+  
                        "Choice 3 : Display\n"+  
                        "Any other choice : Exit\n"+  
                        "Enter your choice\n");  
        ch=sc.nextInt();  
        switch(ch){  
            case 1:  
                System.out.println("Enter the element to be pushed");  
                data=sc.nextInt();  
                s1.push1(data);  
                break;  
            case 2:  
                s1.pop1();  
                break;  
            case 3:  
                 System.out.println("The contents of the stack are  ");  
                s1.display1();  
                break;  
        }  
    }while(ch!=3);  
    }  
}

**2.Question**

**Stack 2 : Array Implementation : Push and Pop**

Consider implementing a dynamically sized stack using an array.

Create a structure

struct stack

{

int \* a;

int top;

int maxSize;

};

The structure stack has a pointer 'a' to a dynamically allocated array (used to hold the contents of the stack), an integer 'maxSize' that holds the size of this array (i.e the maximum number of data that can be held in this array) and an integer 'top' which stores the array index of the top element in the stack.

Write a program to implement push and pop operation on stack and to display the contents of the stack.

In the initstack function intialize the value of top to -1, initialize the value of maxSize and initialize the int array

Print the message “Stack is full” in the push function when an attempt is made to push a data into a full stack.

Print the message “Stack is empty” in the pop function and return the value -1000 when an attempt is made to pop data from an empty stack.

Refer function specifications for further details.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

Note that the statement “The contents of the stack are” is in the main function. In the display function, if the stack is empty, print “ {}”.

[All text in bold corresponds to input and the rest corresponds to output]

**Sample Input and Output:**

Enter the maximum size of the stack

**3**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**1**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**2**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**3**

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**1**

Enter the element to be pushed

**4**

Stack is full

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**3**

The contents of the stack are 1 2 3

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

The popped element is 3

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

The popped element is 2

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

The popped element is 1

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**2**

Stack is empty

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**3**

The contents of the stack are {}

Choice 1 : Push

Choice 2 : Pop

Choice 3 : Display

Any other choice : Exit

Enter your choice

**4**

**Program**

import java.io.\*;  
import java.util.Scanner;  
class Stack1{  
    int top=-1;  
    int size;  
    int[] arr;  
      public Stack1(int size){  
        this.size=size;  
        arr=new int[size];  
         
    }  
    public void push1(int data){  
        if(top==size-1){  
            System.out.println("Stack is Full");  
        }else{  
            top++;  
            arr[top]=data;  
        }  
    }  
    public void pop1(){  
        if(top==-1){  
            System.out.println("Stack is empty");  
        }else{  
            System.out.println("The popped element from stack is"+arr[top]);  
            int temp=top;  
            top--;  
            temp=0;  
        }  
    }  
    public void display1(){  
        if(top==-1){  
            System.out.println("Stack is Empty");  
        }else{  
            for(int i=0;i<=top;i++){  
                 
                System.out.println(arr[i]);  
            }  
        }  
    }  
}  
  
  
  
public class Main{  
    public static void main(String args[]){  
        Scanner sc=new Scanner(System.in);  
         int ch,size,data;  
        System.out.println("Enter the maximum size of the stack");  
       size=sc.nextInt();  
        Stack1 s1=new Stack1(size);  
        do{  
         
       System.out.print("Choice 1 : Push\n"+  
                        "Choice 2 : Pop\n"+  
                        "Choice 3 : Display\n"+  
                        "Any other choice : Exit\n"+  
                        "Enter your choice\n");  
        ch=sc.nextInt();  
        switch(ch){  
            case 1:  
                System.out.println("Enter the element to be pushed ");  
                data=sc.nextInt();  
                s1.push1(data);  
                break;  
            case 2:  
                s1.pop1();  
                break;  
            case 3:  
                 System.out.println("The contents of the stack are: ");  
                s1.display1();  
                break;  
        }  
    }while(ch!=3);  
    }  
}

**3.Question**

Write a menu driven program to implement stack as a linked list. Assume that the size of the stack is 5.  
  
**Define a structure**  
  
struct node {  
    int data ;  
    struct node \* link ;  
} ;  
  
Include functions:  
    **push**--- to push an element into the stack.  
    **pop**--- to pop an element from the stack. If the stack is empty, print *“There is no elements in the stack”*  (without quotes)  
    **display**--- to display the elements in the stack. If empty display **“{}”**(without quotes)  
    **count**--- to count the number of elements in the stack  
  
Refer function specifications for further details.  
  
**Input and Output Format:**  
Get the choice of the function to be executed from the user:  
  
    Choice 1 --- push  
    Choice 2 --- pop  
    Choice 3 --- display  
    Choice 4 --- count  
    Any other choice --- exit  
  
Choice 1: Enter the integer data to be pushed into the stack.  
Choice 2: No additional input  
Choice 3: No additional input  
Choice 4: No additional input  
  
**Sample Input & Output:**  
  
Choice 1 : To push data  
Choice 2 : To pop data  
Choice 3 : To display  
Choice 4 : To count  
  
Enter your choice  
**1**  
Enter the data to be pushed  
**34**  
Enter your choice  
**1**  
Enter the data to be pushed  
**23**  
Enter your choice  
**1**  
Enter the data to be pushed  
**55**  
Enter your choice  
**3**  
Stack Contents are  
55 23 34  
Enter your choice  
**1**  
Enter the data to be pushed  
**89**  
Enter your choice  
**3**  
Stack Contents are  
89 55 23 34  
Enter your choice  
**4**  
Number of elements in the linked list is 4  
Enter your choice  
**2**  
Element Popped is 89  
Enter your choice  
**3**  
Stack Contents are  
55 23 34  
Enter your choice  
**2**  
Element Popped is 55  
Enter your choice  
**2**  
Element Popped is 23  
Enter your choice  
**4**  
Number of elements in the linked list is 1  
Enter your choice  
**2**  
Element Popped is 34  
Enter your choice  
**2**  
There is no elements in the stack  
Enter your choice  
**4**  
Number of elements in the linked list is 0  
Enter your choice  
**3**  
Stack Contents are  
{}  
Enter your choice  
**6**  
END

**Program**

import java.io.\*;

import java.util.Scanner;

class Node{

int data;

Node next;

Node(int data){

this.data=data;

}}

class Stack{

Node top=null;

public void push(int data){

Node n=new Node(data);

n.next=top;

top=n;

}

public void pop(){

System.out.println("Element Popped is "+top.data);

Node temp=top;

top=top.next;

temp=null;

}public void display(){

Node temp=top;

while(temp!=null){

System.out.println(temp.data);

temp=temp.next;

}

}public void count(){

Node temp=top;

int i=0;

while(temp!=null){

i++;

temp=temp.next;

}System.out.println("Number of elements in the linked list is"+i);

}}

public class Main{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int ch,data;

Stack s1=new Stack();

do{

System.out.print("Choice 1 : Push\n"+

"Choice 2 : Pop\n"+

"Choice 3 : Display\n"+

"Choice 4: Count\n"+

"Any other choice : Exit\n"+

"Enter your choice\n");

ch=sc.nextInt();

switch(ch){

case 1:

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

data=sc.nextInt();

s1.push(data);

break;

case 2:

s1.pop();

break;

case 3:

System.out.println("Stack Contents are ");

s1.display();

break;

case 4:

System.out.println("Number of elements in the linked list is:");

s1.count();

break;

}

}while(ch!=3);

}

}