LABSHEET-1 (STACK)

1. Create an Integer array A of size 15. Write functions to perform the following operations on the array: a) Insert element into array next to the last occupied. b) Delete the last inserted element. c) Check if the array is completely occupied. d) Check if the array is empty. e) Display the content of the array Write a main program to insert two more elements into the array A, delete 5 elements, and check the empty and full conditions of the array. After each insertion and deletion operation display the contents of the array.

Sample Array A=[5,10,15,20,25,30,35,40,45,50] After calling function insert(100) a) A=[5,10,15,20,25,30,35,40,45,50,100] After calling function delete() b) A=[5,10,15,20,25,30,35,40,45,50]

**import** java**.**util**.**Scanner**;**

public class Stack **{**

public static void main**()** **{**

int**[]** array **={**5**,**10**,**15**,**20**,**25**,**30**,**35**,**40**,**45**,**50**};**

int top**,**cho**,**i**;**

Scanner scan **=** **new** Scanner**(**System**.**in**);**

top**=**9**;**

**while** **(true)** **{**

System**.**out**.**println**(**"STACK OPERATIONS"**);**

System**.**out**.**println**(**"1.Insert"**);**

System**.**out**.**println**(**"2.Remove"**);**

System**.**out**.**println**(**"3.Check Full"**);**

System**.**out**.**println**(**"4.Check Empty"**);**

System**.**out**.**println**(**"5.Display"**);**

System**.**out**.**println**(**"6.Exit"**);**

System**.**out**.**print**(**"Enter your choice : "**);**

ch **=** scan**.**nextInt**();**

**if** **(**ch **>=** 6**)**

**break;**

**switch(**ch**)** **{**

**case** 1**:** **if(**top **==** 14**)**

System**.**out**.**println**(**"Memory Full"**);**

**else{**

top**++;**

array**[**top**]** **=** scan**.**nextInt**();**

**}**

**break;**

**case** 2**:** **if(**top **==** **-**1**)**

System**.**out**.**println**(**"Nothing to clear!"**);**

**else**

top**--;**

**break;**

**case** 3**:** **if(**top **==** 14**)**

System**.**out**.**println**(**"Full"**);**

**else**

System**.**out**.**println**(**"Not Full"**);**

**break;**

**case** 4**:** **if(**top **==** **-**1**)**

System**.**out**.**println**(**"Empty"**);**

**else**

System**.**out**.**println**(**"Not Empty!"**);**

**break;**

**case** 5**:** **if(**top **==** **-**1**)**

System**.**out**.**println**(**"Empty"**);**

**else** **{**

System**.**out**.**println**();**

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

System**.**out**.**println**(**array**[**i**]** **+** " "**);**

System**.**out**.**println**();**

**}**

**break;**

**}**

**}**

**}**

**}**

2. Create a character array B of size 15. Write functions to perform the following operations on the array a) Insert element into array next to the last occupied. b) Delete the first inserted element. c) Check if the array is completely occupied. d) Check if the array is empty. e) Display the content of the array

Write a main program to insert two more elements into the array B, delete 5 elements, and check the empty and full conditions of the array. After each insertion and deletion operation display the contents of the array

Sample Array B=[A,B,C,D,E,F,G,H,I,J] After calling function insert(P) B=[A,B,C,D,E,F,G,H,I,J,P] After calling function delete b) B=[B,C,D,E,F,G,H,I,J,P]

**import** java**.**util**.**Scanner**;**

public class Stack **{**

public static void main**()** **{**

int top**,** cho**,** i**;**

char**[]** array**={**'A'**,**'B'**,**'C'**,**'D'**,**'E'**,**'F'**,**'G'**,**'H'**,**'I'**,**'J'**};**

top **=** 9**;**

Scanner scan **=** **new** Scanner**(**System**.**in**);**

**while** **(true)** **{**

System**.**out**.**println**(**"STACK OPERATIONS"**);**

System**.**out**.**println**(**"1.Insert"**);**

System**.**out**.**println**(**"2.Remove"**);**

System**.**out**.**println**(**"3.Check Full"**);**

System**.**out**.**println**(**"4.Check Empty"**);**

System**.**out**.**println**(**"5.Display"**);**

System**.**out**.**println**(**"6.Exit"**);**

System**.**out**.**print**(**"Enter your choice : "**);**

cho **=** scan**.**nextInt**();**

**if** **(**cho **==** 6**)**

**break;**

**switch(**cho**)** **{**

**case** 1**:** **if** **(**top **==** 14**)**

System**.**out**.**println**(**"Memory Full"**);**

**else** **{**

top**=**top**+**1**;**

array**[**top**]** **=** scan**.**nextInt**();**

**}**

**break;**

**case** 2**:** **if** **(**top **==** **-**1**)**

System**.**out**.**println**(**"Nothing to clear"**);**

**else**

top**--;**

**break;**

**case** 3**:** **if** **(**top **==** 14**)**

System**.**out**.**println**(**"Full"**);**

**else**

System**.**out**.**println**(**"Not Full!"**);**

**break;**

**case** 4**:** **if** **(**top **==** **-**1**)**

System**.**out**.**println**(**"Empty"**);**

**else**

System**.**out**.**println**(**"Not Empty"**);**

**break;**

**case** 5**:** **if** **(**top **==** **-**1**)**

System**.**out**.**println**(**"Empty\n"**);**

**else** **{**

System**.**out**.**println**(**""**);**

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

System**.**out**.**println**(**array**[**i**]** **+** " "**);**

System**.**out**.**println**(**""**);**

**}**

**}**

**}**

**}**

**}**

LABSHEET-2 (QUEUES)

1. Create a stack data structure and do the following operations push(x), pop(),getSize(),getTop(), isEmpty(), IsFull(), displayElements()

**import** java**.**util**.**Scanner**;**

public class StackDemo **{**

private int n\_size **=** 10**;**

int A**[]** **=** **new** int**[**n\_size**];**

int ch**[]=new** int**[**n\_size**];**

int i**=**0**;**

int top **=** **-**1**;**

public void push**(**int x**)** **{**

// Implement push operation

**if(**top**>=**n\_size**)**

System**.**out**.**println**(**"OverFlow"**);**

//Display proper message

**else**

**{**

top**=**top**+**1**;**

A**[**top**]=**x**;**

**}**

// Call displayElements()

displayElements**();**

**}**

public boolean pop**()** **{**

// Implement pop operation

**if(**top**<**0**)**

**{**

System**.**out**.**println**(**"UnderFlow\n"**);**

**return** **false;**

**}**

**else**

**{**

System**.**out**.**println**(**"\n"**);**

**if(**ch**[**top**]==**'{' **&&** ch**[**i**]==**'}'**)**

top**=**top**-**1**;**

**if(**ch**[**top**]==**'[' **&&** ch**[**i**]==**']'**)**

top**=**top**-**1**;**

**if(**ch**[**top**]==**'(' **&&** ch**[**i**]==**')'**)**

top**=**top**-**1**;**

**else**

**return** **false;**

**}**

//Display proper message

displayElements**();**

// Call displayElements()

**}**

public void displayElements**()** **{**

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

**for** **(**int i **=** 0**;** i **<=** top**;** i**++)** **{**

System**.**out**.**print**(**A**[**i**]+**"\t"**);**

**}**

**}**

public int getSize**()** **{return** n\_size**;}**

public int getTop**()** **{** **return** top**;**

**}**

public boolean isEmpty**(){**

**if(**top**<**0**)**

**return** **true;**

**else**

**return** **false;**

**}**

public boolean isFull**(){**

**if(**top**>=**n\_size**)**

**return** **true;**

**else**

**return** **false;**

**}**

public boolean Brackets**()** **{**

Scanner sc**=new** Scanner**(**System**.**in**);**

**for(**int k**=**0**;**k**<**s**.**getSize**();**k**++){**

int temp**=**sc**.**nextChar**();**

**if(**temp**==**'{' **||** temp**==**'[' **||** temp**==**'('**)**

**{**i**++;**s**.**push**(**temp**);}**

**else**

**if(**s**.**pop**()==false)**

**return** **false;**

**else**

**return** **true;**

**}**

**}**

public static void main**(**String**[]** args**)** **{**

StackDemo s **=** **new** StackDemo**();**

Scanner sc**=new** Scanner**(**System**.**in**);**

**for(**int i**=**0**;**i**<**s**.**getSize**();**i**++){**

s**.**push**(**sc**.**nextInt**());**

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

**}**

//Try push on a full stack

**for(**int i**=**s**.**getTop**();**i**>=**0**;**i**--)**

**if(**s**.**pop**());**

//Try pop on empty stack

int n**=** s**.**getSize**();**

System**.**out**.**println**(**n**);**

//print n

int m**=** s**.**getTop**();**

System**.**out**.**println**(**m**);**

**if(**s**.**Brackets**()==true)**

System**.**out**.**println**(**"No errors"**);**

**else**

System**.**out**.**println**(**"Errors Detected"**);**

sc**.**close**();**

**}**

**}**

Implement the balanced parenthesis algorithm discussed in class as a function and add it into previous code. Input following expressions and check whether it is balanced or not.

{ ( ) { [ ( ) ] } }

[ ( ] )

) (

{ ( ) ( ) ( ) [ { } ] }