```
import java.util.Scanner;

public class STACK_ARRAY_DEMO
{
        static int MAX=10;

        static int TOP = -1 ;

        static int stack[]=new int[MAX];

        public static void main(String[] args)
        {
                Scanner sc=new Scanner(System.in);

                char ch;
                int opt;
                int item;

                do
                {       System.out.println("1.PUSH 2.POP 3.DISPLAY/TRAVERSE 4.PEEK ");

                        System.out.println("Enter your option");
                        opt=sc.nextInt();

                        switch(opt)
                        {
                                case 1:  System.out.println("Enter the item to push:");
                                         item=sc.nextInt();
                                         PUSH(item);
                                         break;

                                case 2:  POP();
                                         break;

                                case 3:  TRAVERSE_STACK();
                                         break;

                                case 4:  PEEK();
                                         break;
                                default:
                                         System.out.println("invalid option");
                        }  /* End of switch */
```

```
                System.out.println("\nDo you want to perform another operation(y/n)");
                ch=sc.next().charAt(0);

            }while(ch=='y'|| ch== 'Y'); /*End of do…while loop*/

        } /* End of main method */

/*isFull() method to check overflow condition in java*/

    public static boolean isFull()
    {
        if ( TOP >= MAX - 1 )
            return true;
        else
            return false;
     }
    /*End of isFull method*/

/*isEmpty() method to check underflow condition in java*/
    public static boolean isEmpty()
     {
        if ( TOP <= -1 )
            return true;
        else
            return false;
     }
 /*End of isEmpty method*/

/*PUSH() method to insert a new data item at top of the stack*/

    public static void PUSH(int new_item)
     {
        if ( isFull() == true )
        {
            System.out.println("The stack is full");
            System.out.println("You can't insert more items");
            return;
        }
        else
```

```
            {       TOP = TOP + 1;
                    stack[TOP] = new_item;
            }
        }
/*End of  PUSH() method */


/*POP() method to delete the topmost data item from the stack*/

        public static void POP()
        {
            if ( isEmpty() == true )
            {
                    System.out.println("The stack is empty…you can't delete an item");
                    return;
            }
            else
            {       int temp = stack[TOP];
                    TOP = TOP - 1;
                    System.out.print("\n The popped item is : " + temp );
            }
        }
/*End of POP() method */


/*PEEK() method to display or return the topmost data item in the stack*/

        public static void PEEK()
        {
            if ( isEmpty() == true )
            {
                    System.out.println("The stack is empty…you can't peek");
                    return;
            }
            else
            {       int temp = stack[TOP];
                    System.out.print("\n The peeked item is : " + temp );
            }
         }
/*End of PEEK() method */
```

```
/*TRAVERSE_STACK() method to display the stack elements from top to bottom*/

        public static void TRAVERSE_STACK()
        {
                if ( isEmpty() == true )
                {
                        System.out.println("The stack is empty…you can't traverse");
                        return;
                }
                else
                {   System.out.println("The stack elements from top to bottom are");
                    int i = TOP ;
                    while ( i >= 0 )
                    {
                        System.out.print(stack[i] + " → " );
                        i = i - 1;
                     }
                }
        }
/*End of TRAVERSE_STACK() method */

}  /* END OF STACK_ARRAY_DEMO CLASS */
```