

<https://course.acciojob.com/idle?question=6741eaff-749d-4bef-a8c9-7c768de975a0>

● EASY

● Max Score: 30 Points

●

●

●

## Design Stack Using Array

Your task is to implement a stack using an array. You will be given  $Q$  queries in an input array `queries`. In the driver code you will be given an empty array named `st`. You just need to complete the function of the given class methods.

These are the required functions in the stack:

1. `push()`: Inserts a value in the stack.
2. `pop()`: Removes a value from the top of the stack.
3. `peek()`: Returns the top value of the stack.
4. `display()`: Print the items in the stack.

### Note

If the stack is empty and the peek function is called return -1.

## Input Format

You are given an empty array `st` and you need to implement a stack.

The input is handled by the driver code.

In driver code, input is taken as:

First line contains an integer  $Q$  denoting the number of operations.

Next  $Q$  lines could contain one of four numbers 1, 2, 3, 4.

**1 denotes the insertion operation and is followed by the value to be inserted.**

**2 denotes peek operation**

**3 denotes the pop operation**

**4 denotes display operation**

## Output Format

**Complete the given functions.**

### Example 1

Input

```
6
1 5
1 9
2
3
3
2
```

Output

```
9
-1
```

Explanation

First, we enter 5 and 9 into the stack. Now we check the top of the stack which is 9. Therefore 9 is printed. Now we pop two values from the stack and then again check the top of the stack. As the stack is empty this time -1 is printed.

### Example 2

Input

```
6
1 5
```

```
1 9
1 3
3
2
4
```

Output

```
9
9 5
```

Explanation

First, we enter 5, 9, and 3 into the stack. Now we pop once. We now check the top element which is 9. Therefore 9 is printed. Now we display the list which is 9 5.

## Constraints

$1 \leq Q \leq 10000$

$1 \leq \text{value that can be added in stack} \leq 10000$

### Topic Tags

- **Stacks**
- **Arrays**

# My code

// in java

```
import java.util.*;
```

```
class Main {
    public static void main(String[] args) {
        StackUsingLinkedlist obj = new StackUsingLinkedlist();
```

```

Scanner sc = new Scanner(System.in);
int q;
q = sc.nextInt();
while (q-- > 0) {
    int x;
    x = sc.nextInt();
    if (x == 1) {
        int y;
        y = sc.nextInt();
        obj.push(y);
    }
    if (x == 2) {
        System.out.println(obj.peek());
    }
    if (x == 3) {
        obj.pop();
    }
    if (x == 4) {
        obj.display();
    }
}
sc.close();
}
}

```

```

// class StackUsingLinkedlist {

//     ArrayList<Integer> st;
//     int top=-1;
//     StackUsingLinkedlist() {
//         //Complete the function
//         st=new ArrayList<Integer>();
//     }
// }

```

```

//    }

//    public void push(int x) {
//        //Complete the function
//        st.add(++top,x);

//    }

//    public int peek() {
//        //Complete the function
//        if(st.size()==0)
//            return -1;
//        return st.get(top);
//    }

//    public void pop() {
//        //Complete the function
//        st.remove(top);
//        top--;

//    }

//    public void display() {
//        //Complete the function
//        for(int i=top;i>=0;i--)
//            System.out.print(st.get(i)+" ");
//    }
// }
class StackUsingLinkedlist {

    int st[]=new int[10010];
    int top=-1;

```

```

StackUsingLinkedlist() {
    top=-1;
    for(int i=0; i<10001; i++)
    {
        st[i]=0;
    }
}

public void push(int x) {
    top++;
    st[top]=x;    // st[0]=5,top=0;
}

public int peek() {
    if(top== -1)
    {
        return -1;
    }
    return st[top];
}

public void pop() {
    if(top!= -1)
        top--;
    //Complete the function
}

public void display() {
    for(int i=top; i>=0; i--)
    {
        System.out.print(st[i]+" ");
    }
}

```

```
System.out.println();
```

```
}
```

```
}
```