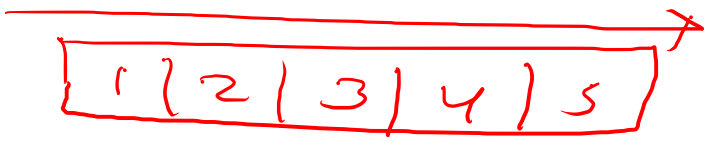
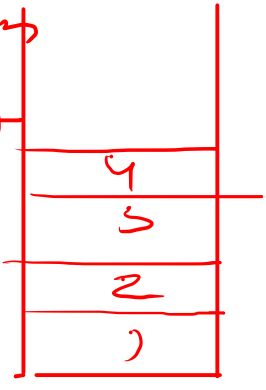
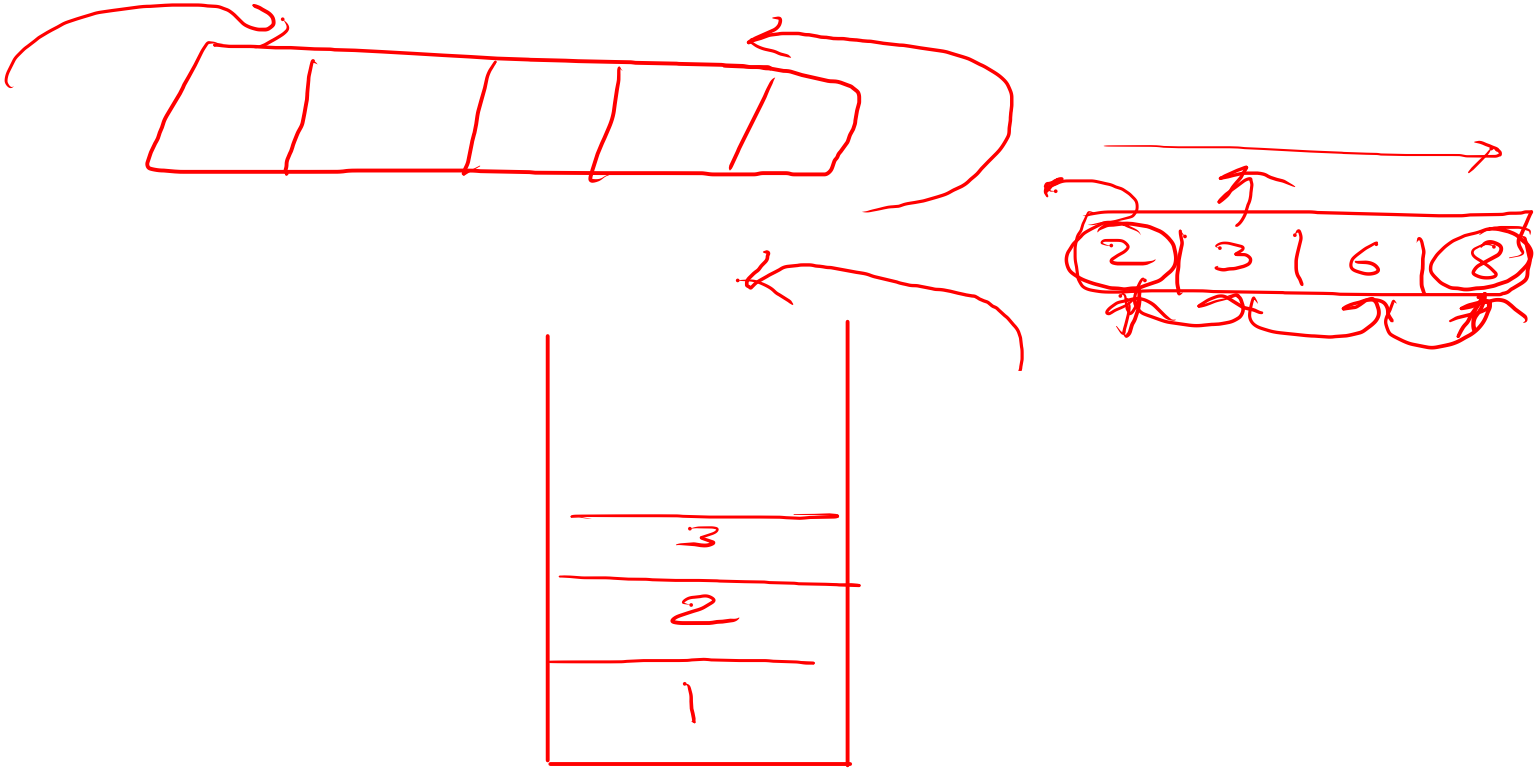
Stack

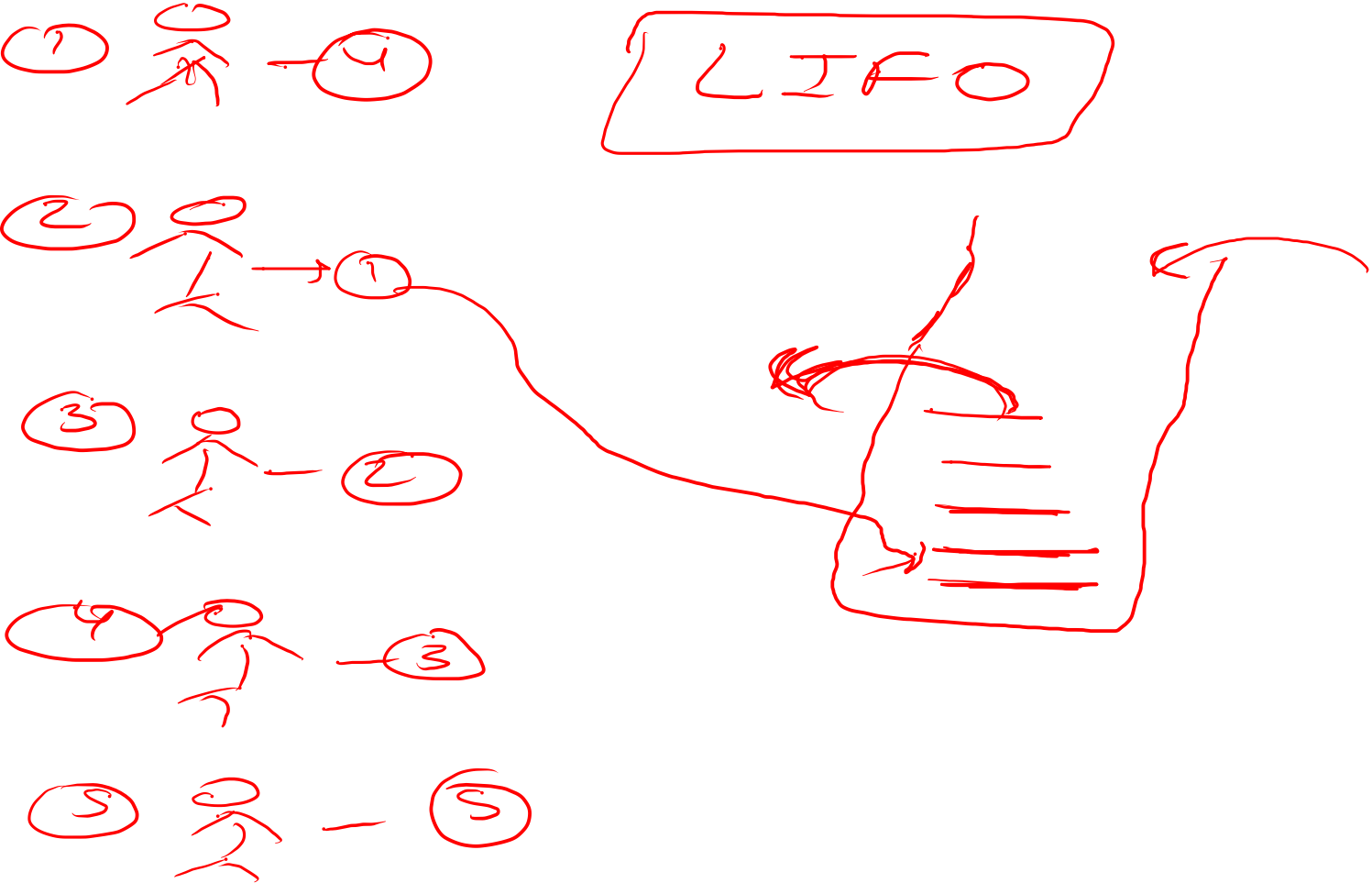
A stack is almost an array.



- Stack is a linear data structure
- Stack is a collection of homogenious element

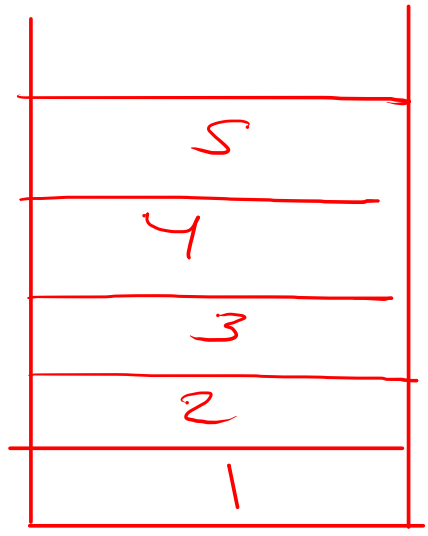



2
3
1
6
8
3
2
1


LIFO

① Hello ② ~~World~~

→ Stack



① Iterator

1 2 3 4 5

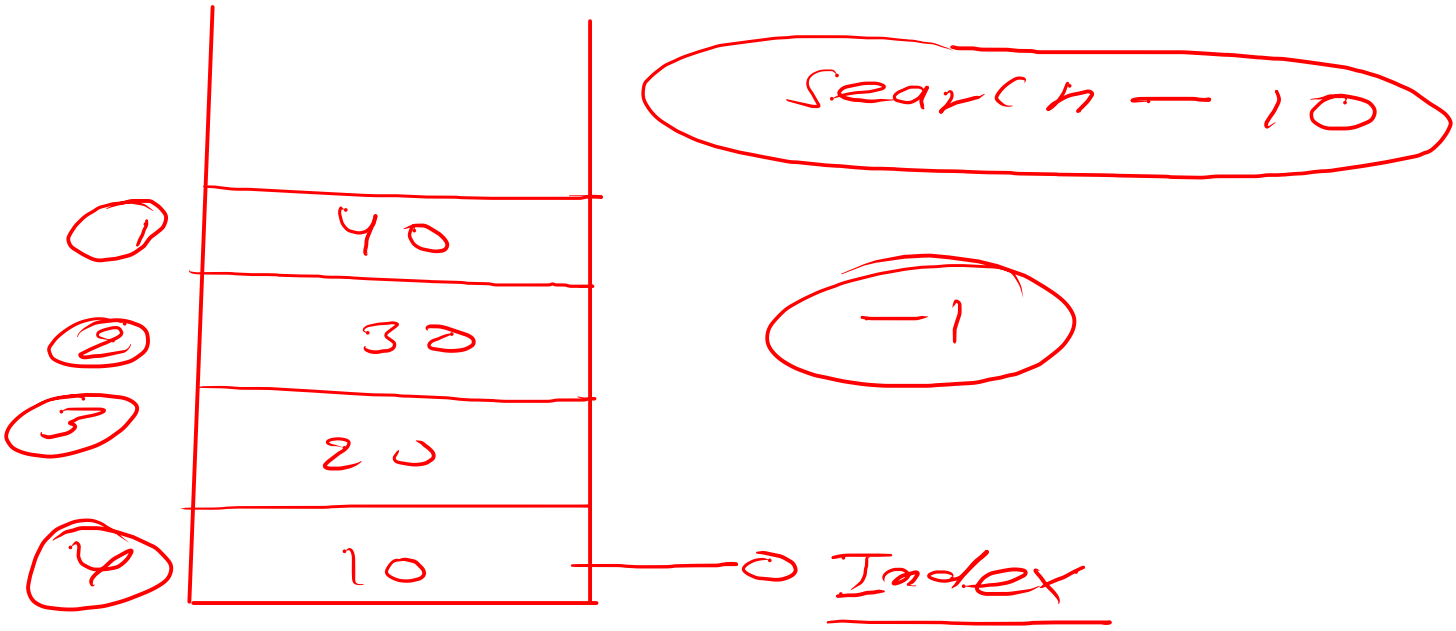
Stack Class

① It is also available in util package.

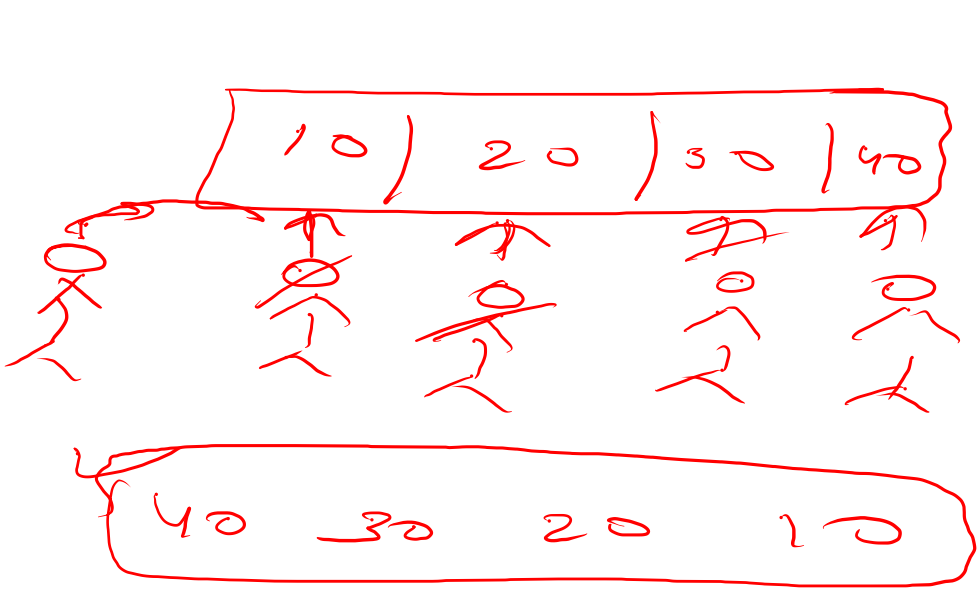
Declaration

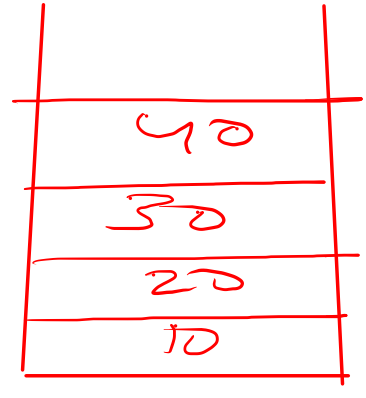
```
Stack<className> objName = new Stack<>();
```

1.Push(): Insert the value at the top.
2.pop(): It will delete the element and return it
3peek(): I will return the top element availabel in the stack
4size(): It will return the size.
5search():It will search the element in the stack.if it is present then it will return the position from the top and it returns -1 if element is not present.


search — 10
1
40
2
30
3
20
4
10
0
Index
— 1

```java
// Iterator
Iterator<Integer> it = st.iterator();
while (it.hasNext()){
    System.out.print(it.next()+" ");
}
```





Stack Syntax Learning

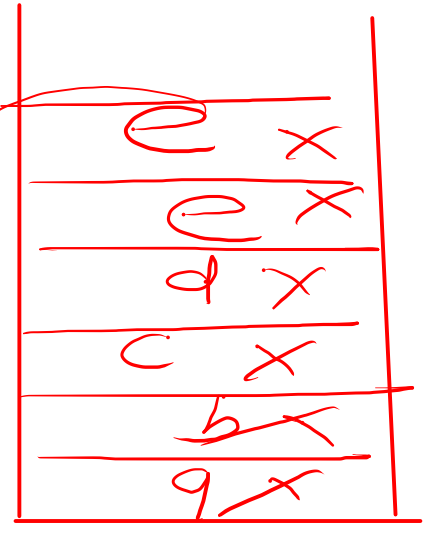
```
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution
       Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        Stack<Integer> st = new Stack<>();
        while(t-->0){
            int val = sc.nextInt();
            if(val==1){
                System.out.println(st.size());
            }else if(val==2){
               if(st.size()==0){
                    System.out.println(-1);
               } else{
                    st.pop();
               }
            }else if(val==3){
                int x = sc.nextInt();
                st.push(x);

            }else if(val==4){
                if(st.size()==0){
                   System.out.println(-1);
               } else{
                  System.out.println(st.peek());
               }
            }
        }
```

for(i=0; i<n

abcdee

Sample Output 0

eedcba

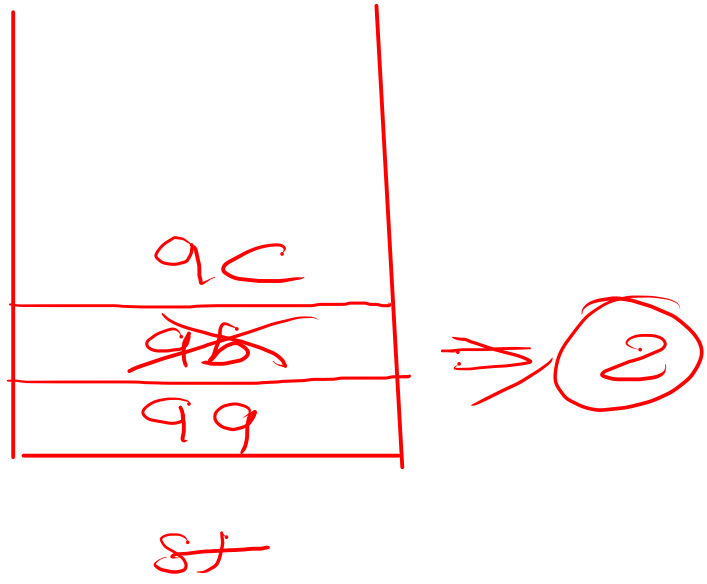


Reverse string

Language: Java 7

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solutic
        Scanner sc = new Scanner(System.in);
        String str  = sc.next();
        reverse(str);
    }
    public static void reverse(String str){
        Stack<Character> st = new Stack<>();
        for(int i=0;i<str.length();i++){
            char ch = str.charAt(i);
            st.push(ch);
        }
        // If we want to print the elements from the stack
        while(!st.isEmpty()){
            System.out.print(st.pop());
        }
    }
}
```

```
4
aa ab ab ac
```



Delete consecutive

Language: Java 7

Open in editor

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        Stack<String> st = new Stack<>();
        for(int i=0;i<n;i++){
            String str = sc.next();
            if(st.search(str)==-1){
                st.push(str);
            }else{
                st.pop();
            }
        }
        System.out.println(st.size());
    }
}
```