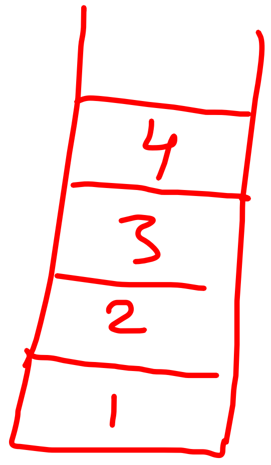
Stack : LIFO (Last In First Out)



Insert/Delete : only top end.

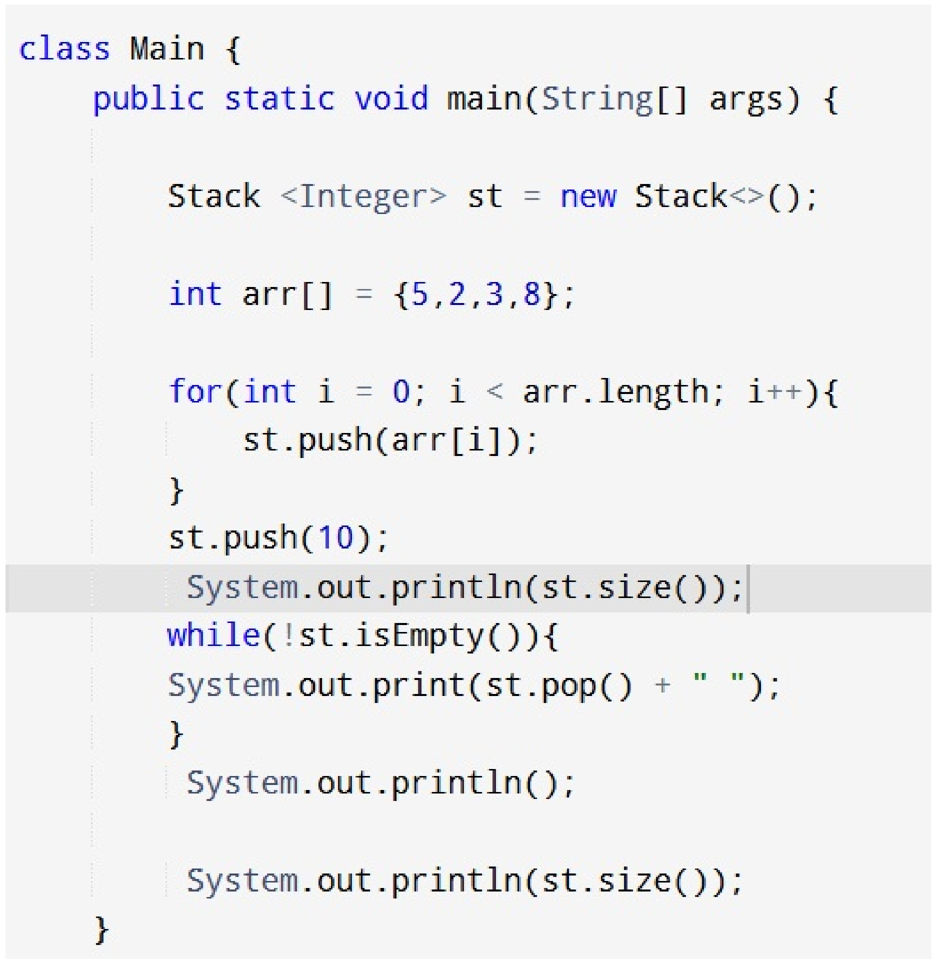
1. Insert → Top → Push O(1)
2. Delete → Top → Pop O(1) → print & remove the top elem
3. Top() → element present at top → Peek O(1) → print top elem
4. isEmpty() → stack is empty or not O(1)
5. Size() → size of the stack O(1)

Stack <Generic/ DataType/ wrapper class > name of stack = new Stack <>();

Stack <Integer> stack = new Stack <>();

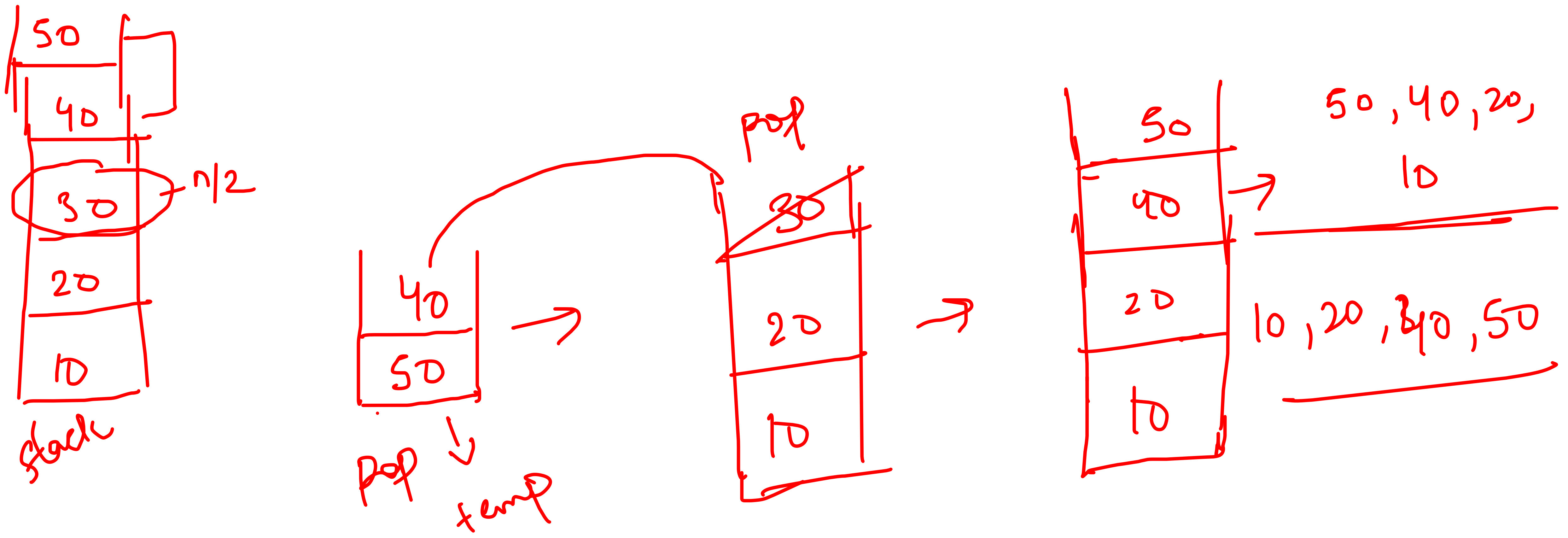
→ StackFullException

→ StackEmptyException



```java
class Main {
    public static void main(String[] args) {

        Stack <Integer> st = new Stack<>();

        int arr[] = {5,2,3,8};

        for(int i = 0; i < arr.length; i++){
            st.push(arr[i]);
        }
        st.push(10);
         System.out.println(st.size());
        while(!st.isEmpty()){
        System.out.print(st.pop() + " ");
        }
         System.out.println();

         System.out.println(st.size());
    }
```

HW_Delete middle element of a stack

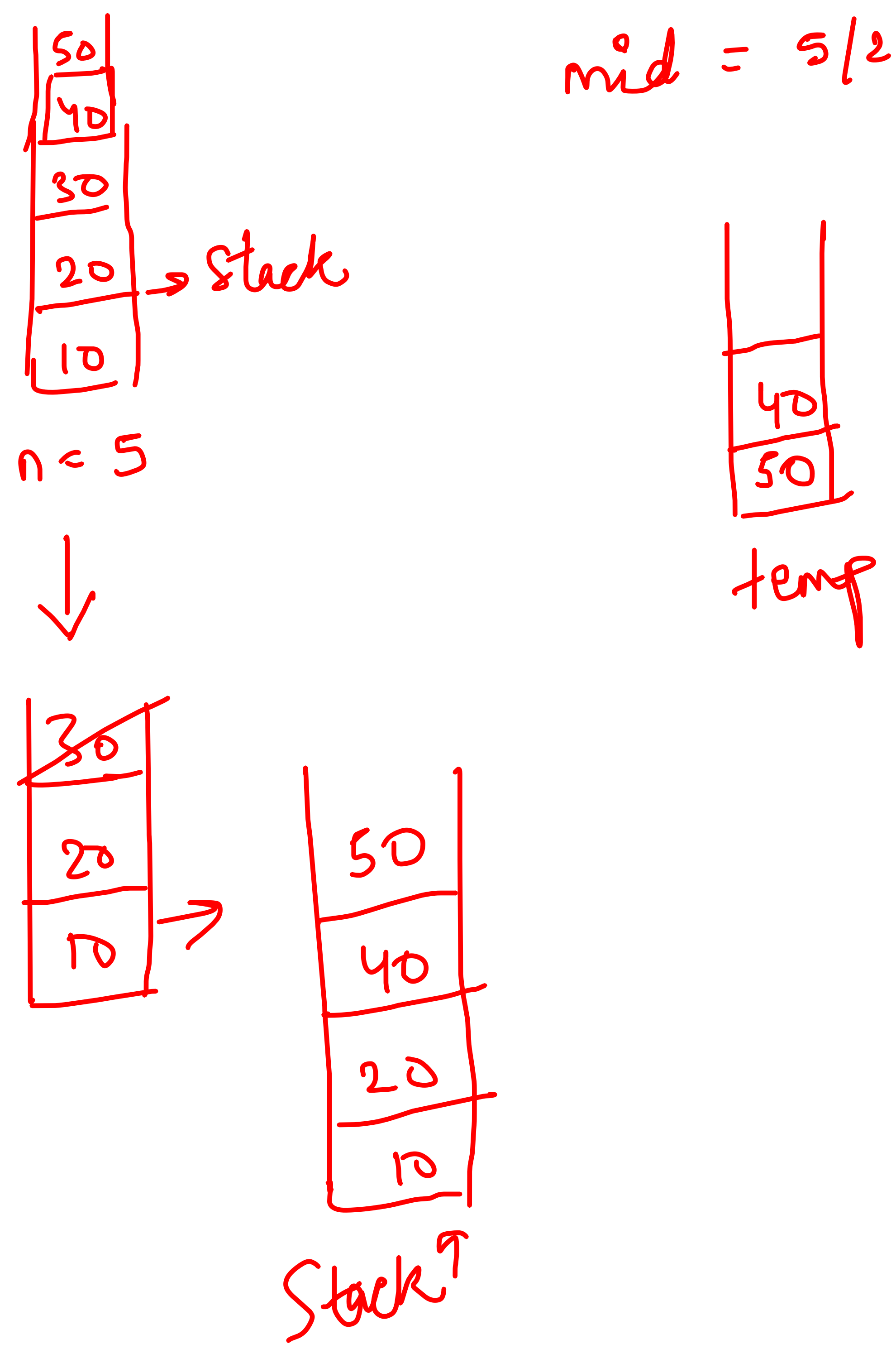


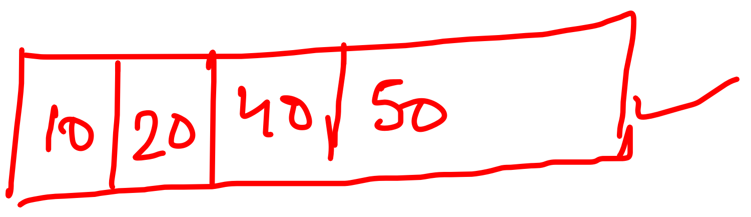
```
Scanner s = new Scanner(System.in);
int n = s.nextInt();

Stack<Integer> stack = new Stack<>();

for(int i = 0; i < n; i++){
    stack.push(s.nextInt());
}

// middle elem
int mid  = n/2;

Stack<Integer> temp = new Stack<>();

for(int i = 0; i < mid; i++){
    temp.push(stack.pop());
}

// remove the mid
stack.pop();

while(!temp.isEmpty()){
    stack.push(temp.pop());
}

List<Integer> elements = new ArrayList<>(stack);
for(int i : elements){
    System.out.print(i + " ");
}
```

$TC \rightarrow O(n)$

$SC \rightarrow O(n)$

mid = 5/2 = 2





o/p → 10 20 40 50

HW_Reversing the equation 1

String Builder $\longrightarrow$

$\downarrow$ mutable

```
20-3+5*2
```

```
Scanner s = new Scanner(System.in);
String eq = s.nextLine();

Stack<String> st = new Stack<>();
StringBuilder number = new StringBuilder();

for(int i = 0; i < eq.length(); i++){
    char c = eq.charAt(i);

        if(Character.isDigit(c)){
            number.append(c);   // number form
        }
    else{
        if(number.length() > 0){
            st.push(number.toString()); // number push
            number.setLength(0);
        }
        st.push(Character.toString(c)); // operator push
    }
}
```

```
// for the remaining number
if(number.length() > 0){
  st.push(number.toString());
}

StringBuilder result = new StringBuilder();

while(!st.isEmpty()){
    result.append(st.pop());
}

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

0 1 2 3 4 5 6 7
20 - 3 + 5 * 2

result = 2 * 5 + 3 - 20.

St = "20" - 3 + 5 * 2

[illegible]

number = ~~20~~

i = 0, c = 2 Yes

i = 1, c = 0 yes

i = 2, c = - no

number > 0 T

i = 3, c = 3 yes

```java
    Scanner s = new Scanner(System.in);
    int n = s.nextInt();
    s.nextLine();

    for(int i = 0; i <n;i++){
        String number = s.nextLine();

    Stack<Character> st = new Stack<>();

    // push each char of the number onto the stack

    for(char c : number.toCharArray()){
        st.push(c);
    }
    StringBuilder reverse = new StringBuilder();
    while(!st.isEmpty()){
        reverse.append(st.pop());
    }

    // convert to string
    // remove leading zeros

    String str = reverse.toString();

    String result = str.replaceFirst("^0+(?!$)", "");

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