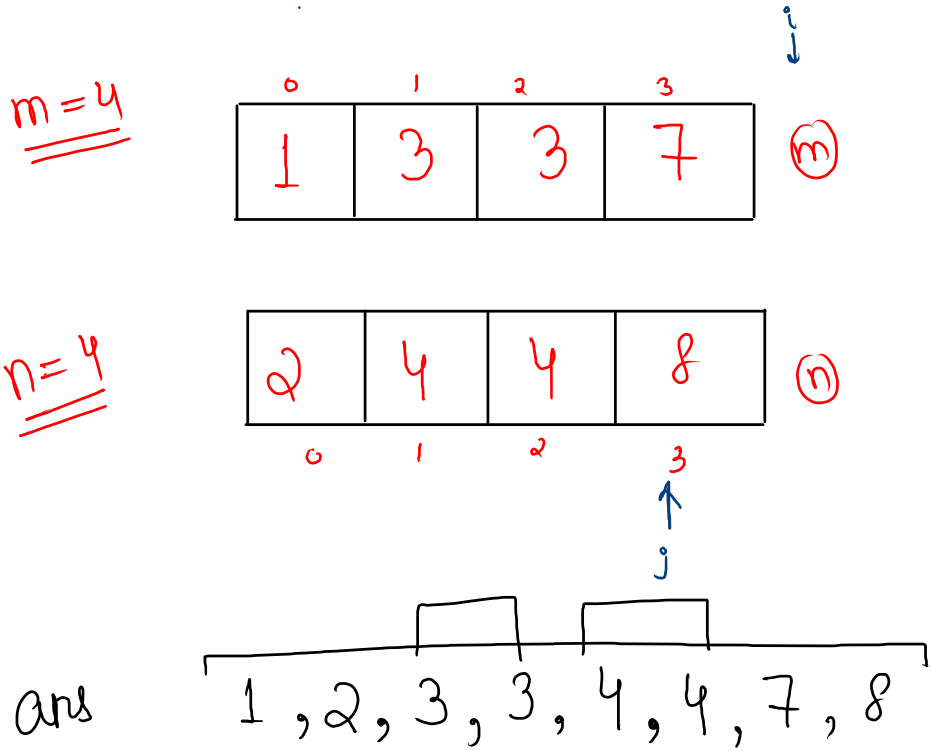
Merge two sorted arrays 7



psudo code

1) int i=0, j=0

2) traverse until i<m && j<n

2.1) compare arr1[i] and arr2[j]
arr1[i] < arr2[j]
ans.add(arr1[i]); i++;

2.2) ans.add(arr2[j]); j++;

3) while(j < n){
ans.add(arr2[j]);
j++;
}

4) while (i < m){
ans.add(arr1[i]);
i++;
}

handling duplicacy

ans :- [1, 2, 3_a, 3_b, 4, 4, 7, 8]

i

0 1 2 3 4 5 6 7

ans.remove(2) [1, 2, 3_b, 4_a, 4_b, 7, 8]

0 1 2 3 4 5 6

ans.remove(3) [1, 2, 3, 4_b, 7, 8]

Code

```java
public static void mergeArrays(int[] arr1, int m, int[] arr2, int n) {
    ArrayList<Integer> ans = new ArrayList<>();
    int i = 0;
    int j = 0;
    while ( j < n && i < m ) {
        if ( arr1[i] < arr2[j] ) {
            ans.add( arr1[i] );
            i++;
        } else {
            ans.add( arr2[j] );
            j++;
        }
    }

    while ( i < m ) {
        ans.add( arr1[i] );
        i++;
    }

    while ( j < n ) {
        ans.add( arr2[j] );
        j++;
    }

    int idx = 0;
    while ( idx < ans.size() - 1 ) {
        if ( ans.get(idx) == ans.get(idx + 1) ) {
            ans.remove(idx);
        } else {
            idx++;
        }
    }

    for ( int k = 0; k < ans.size(); k++ ) {
        System.out.print( ans.get(k) + " " );
    }
}
```

until both pointer are with in array

for remaining element

to handle duplicacy

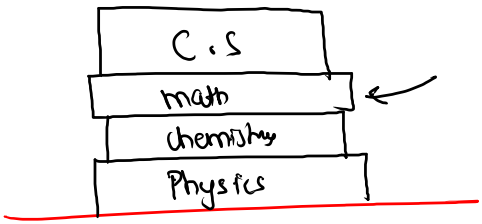
for printing

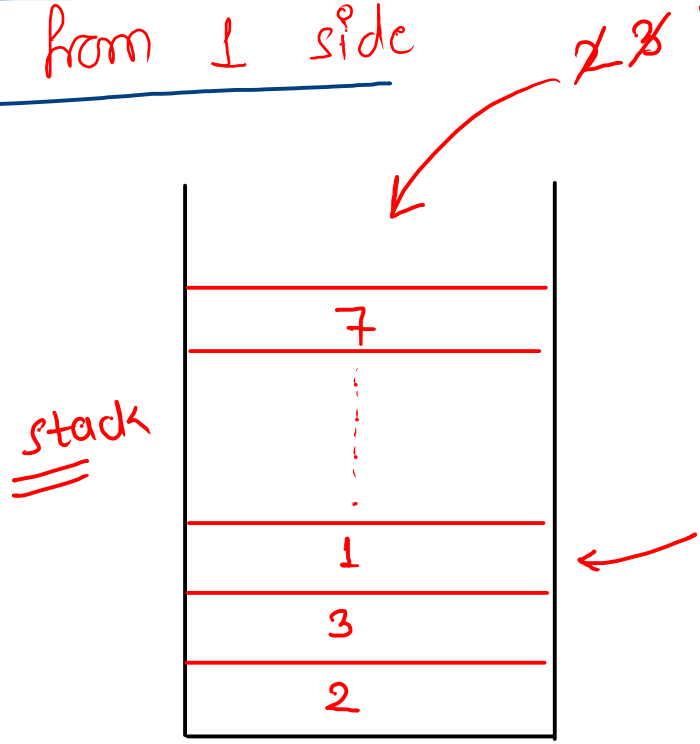
⇒ Stack

(LIFO)
↳ last in first out

↳ same as arraylist but we can only add and remove element from 1 side

restriction





stack

⇒ In-built Functions

syntax

```
Stack< DataType> st = new Stack<>();
```

↳ Integer, Boolean, ------

To add an element in stack : st.push(element)
To remove an element in stack : st.pop()
To get top element in stack : st.peek()
To get size of stack : st.size()
To check for empty stack : st.isEmpty()

Code

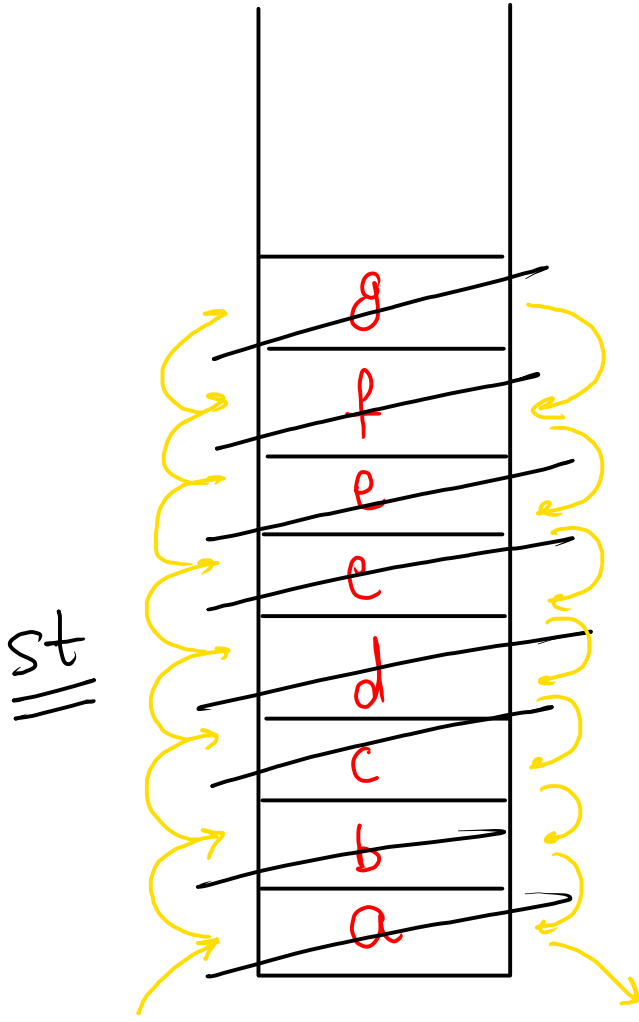
Stack Syntax Learning

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    Stack<Integer> st = new Stack<>();
    int t = scn.nextInt();
    for (int i = 0; i < t; i++) {
        int c = scn.nextInt();
        if ( c == 1 ) {
            printSize(st);
        } else if ( c == 2 ) {
            removeElement(st);
        } else if ( c == 3 ) {
            int x = scn.nextInt();
            addElement(st, x);
        } else if ( c == 4 ) {
            printTopElement(st);
        }
    }
}
public static void printSize(Stack<Integer> st) {
    System.out.println( st.size() );
}
public static void removeElement(Stack<Integer> st) {
    if ( st.isEmpty() ) {
        System.out.println("-1");
        return;
    }
    st.pop();
}
public static void addElement(Stack<Integer> st, int x) {
    st.push(x);
}
public static void printTopElement(Stack<Integer> st) {
    if ( st.isEmpty() ) {
        System.out.println("-1");
        return;
    }
    System.out.println( st.peek() );
}
```

Reverse string

str = "abcdeefg"

ans = "gfeedcba"



code

Reverse string

```java
public class Solution {

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

        Stack<Character> st = new Stack<>();
        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            st.push( ch );
        }

        String ans = "";
        while ( st.size() > 0 ) {
            char top = st.peek();
            ans += top;

            st.pop();
        }

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