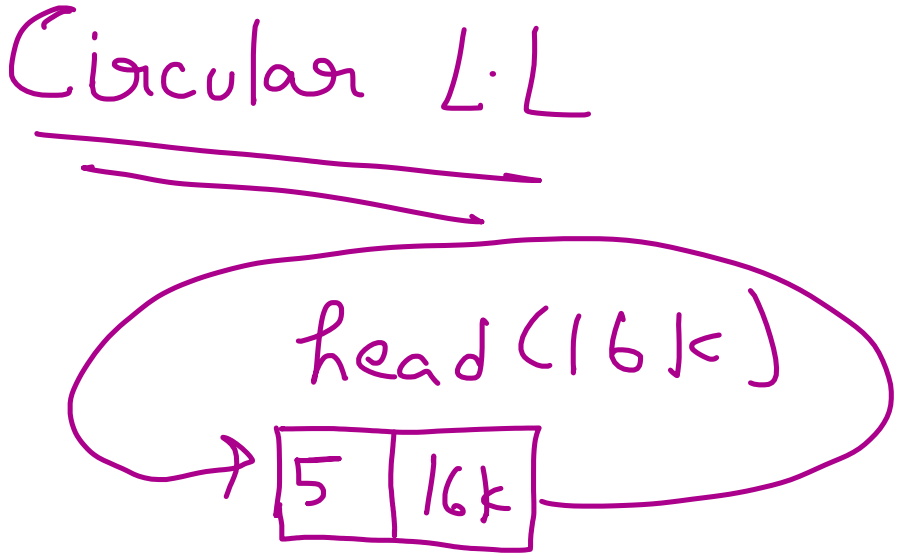
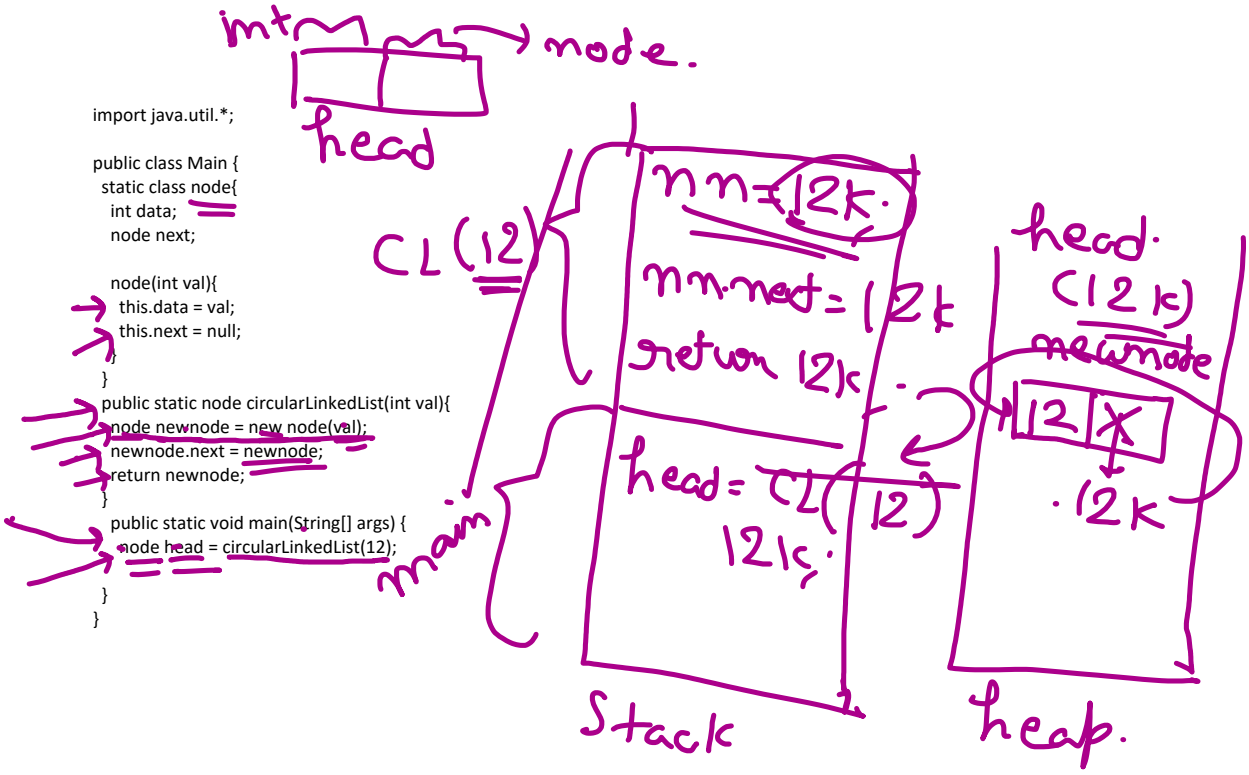
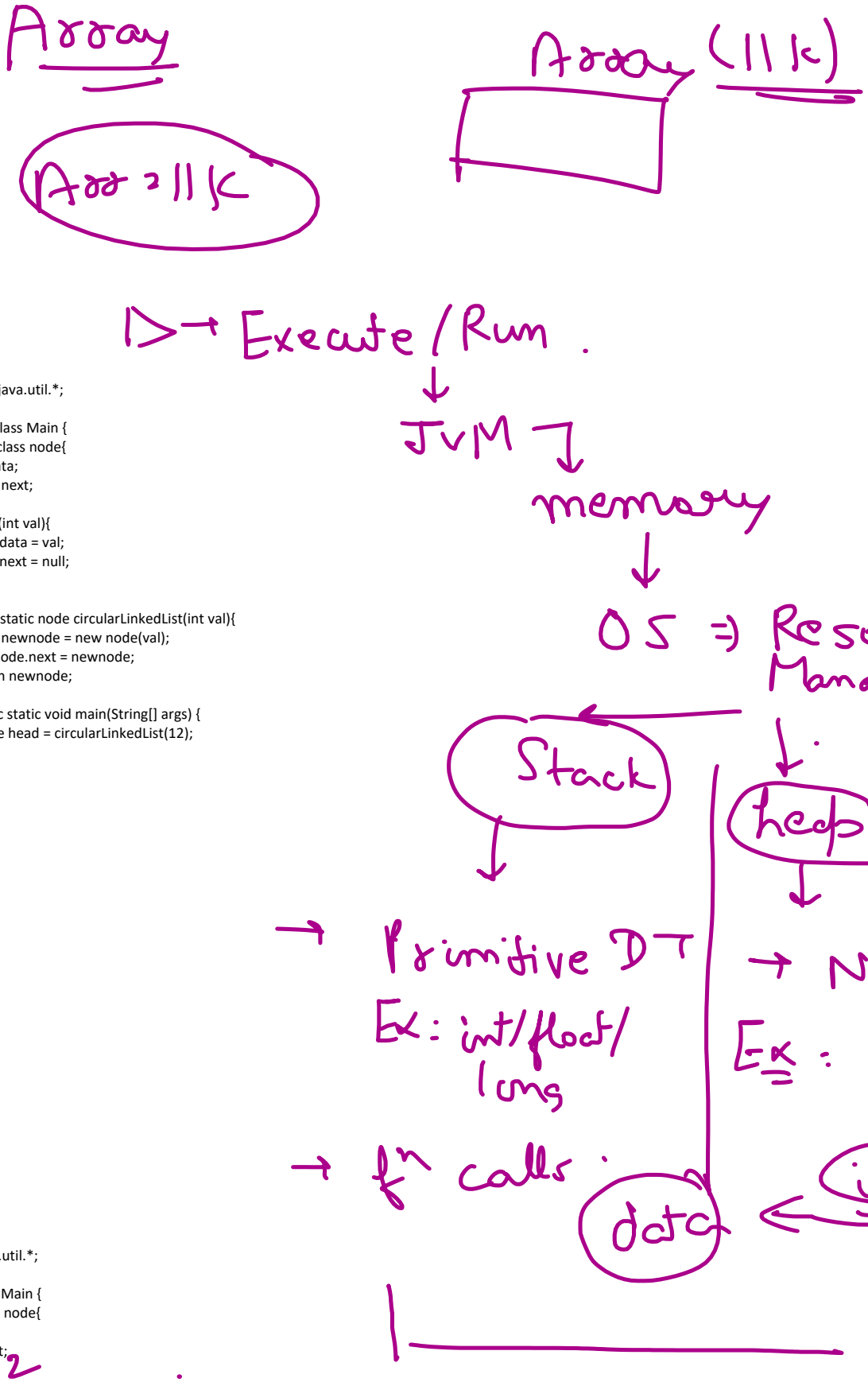
16 February 2024 14:20




```
import java.util.*;

public class Main {
 static class node{
   int data;
   node next;

   node(int val){
    this.data = val;
    this.next = null;
   }
 }
 public static node circularLinkedList(int val){
   node newnode = new node(val);
   newnode.next = newnode;
   return newnode;
 }
   public static void main(String[] args) {
    node head = circularLinkedList(12);
   }
}
```



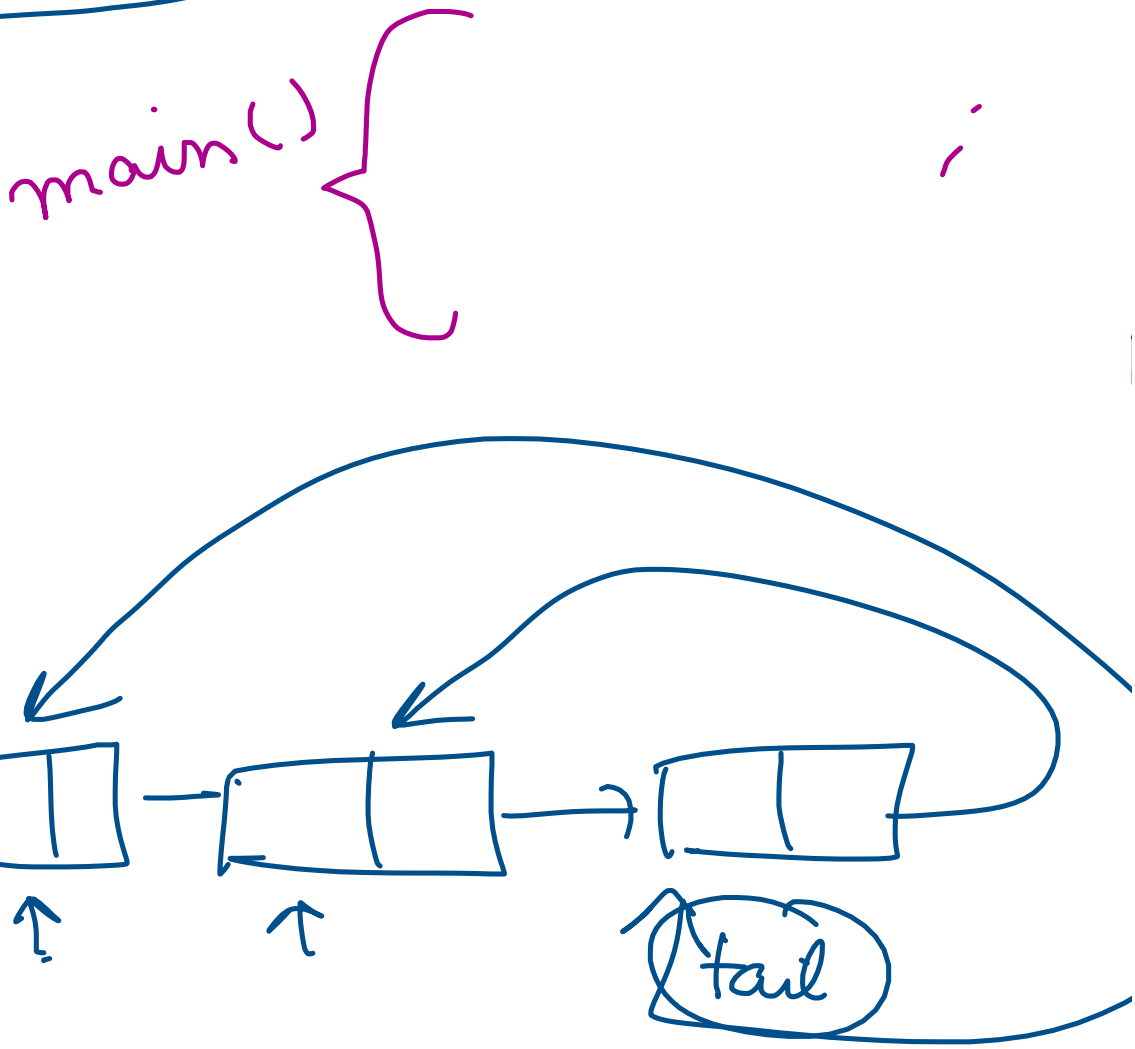



```
import java.util.*;

public class Main {
  static class node{
    int data;
    node next;

    node(int val){
      this.data = val;
      this.next = null;
    }
  }
  public static node circularLinkedList(int val){
    node newnode = new node(val);
    newnode.next = newnode;
    return newnode;
  }
    public static void main(String[] args) {
      node head = circularLinkedList(12);

  }
}
```

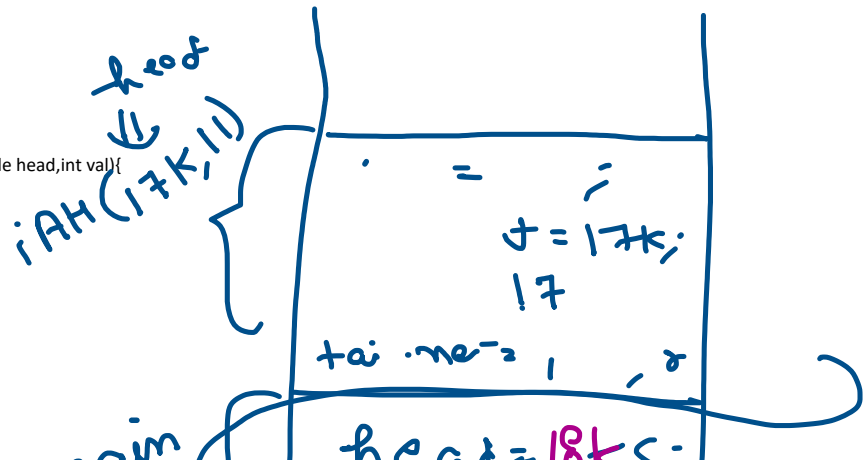
i

```
import java.util.*;

public class Main {
  static class node{
    int data;
    node next;
    node(int val){
      this.data = val;
      this.next = null;
    }
  }
  public static node circularLinkedList(int val){
    node newnode = new node(val);
    newnode.next = newnode;
    return newnode;
  }
```

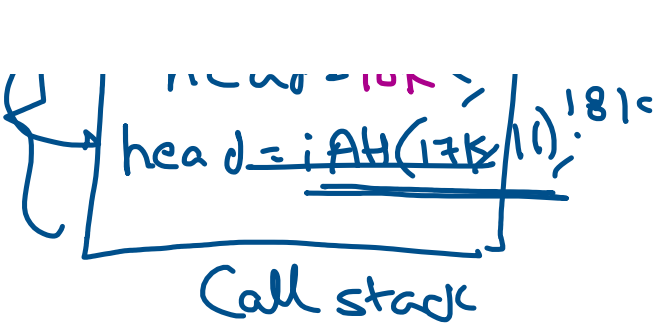
```
    public static void main(String[] args) {
      node head = circularLinkedList(12);
    }
  }
```

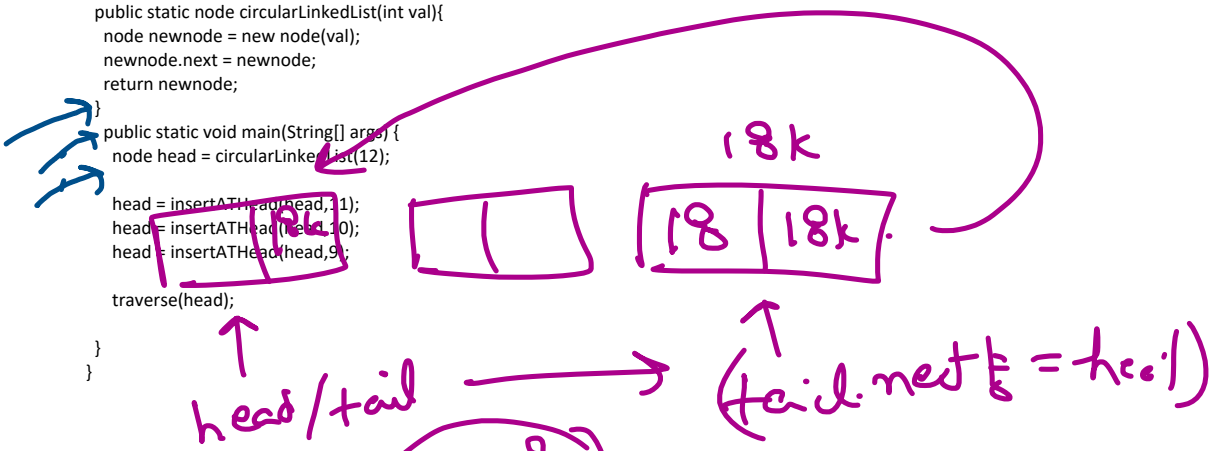


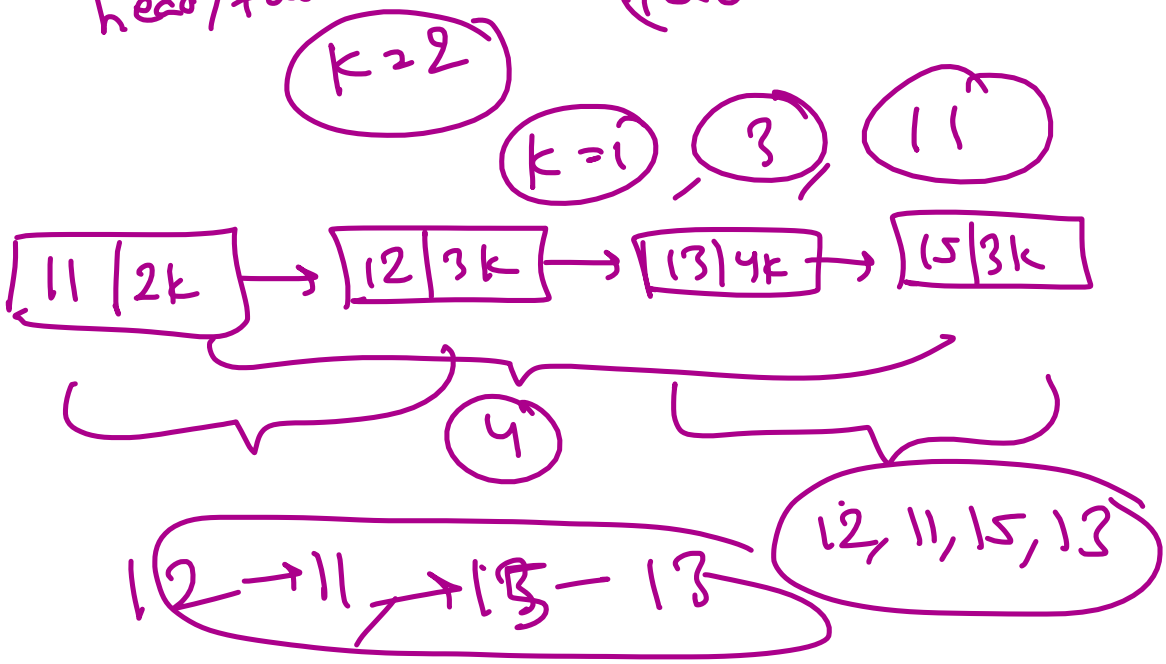

```
import java.util.*;

public class Main {
  static class node{
    int data;
    node next;

    node(int val){
      this.data = val;
      this.next = null;
    }
  }
  public static node insertATHead(node head,int val){
    node newnode = new node(val);
    if(head == null){
      newnode.next = newnode;
      return newnode;
    }
    newnode.next = head;
    node tail = head;
    while(tail.next != head){
      tail = tail.next;
    }
    tail.next = newnode;
```



```
    return newnode;

  }

  public static void traverse(node head){
    if(head == null){
      System.out.print("Circular Linked List is Empty");
      return null;
    }
    node temp = head;
    do{
      System.out.print(temp.data +"->");
      temp = temp.next;
    }
    while(temp != head);
  }
  public static node circularLinkedList(int val){
    node newnode = new node(val);
    newnode.next = newnode;
    return newnode;
  }
    public static void main(String[] args) {
      node head = circularLinkedList(12);

      head = insertATHead(head,11);
      head = insertATHead(head,10);
      head = insertATHead(head,9);

      traverse(head);

    }
  }
```







STACK → push/pop/peek/Empty.

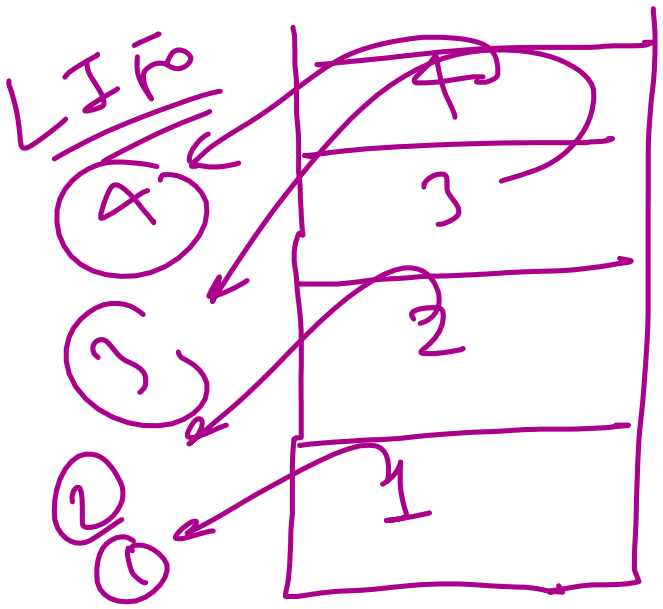
D.S

(LIFO)

Stack →

Linear D.S

↳ (LIFO order)



(LIFO)

← Container.

{1, 2, 3, 4}

gfg ⇒ ① Landing Page / Home

↓

DSA

↓

Stack DS

↓

Problems

push / pop / peek / isEmpty()

~~Array List~~

→ L.L

⇒ ~~Collection framework.~~

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

DT → Integer

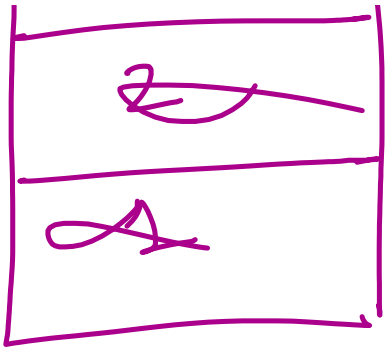
isEmpty()

T F

st.push(1);

st.push(2);

st.pop()



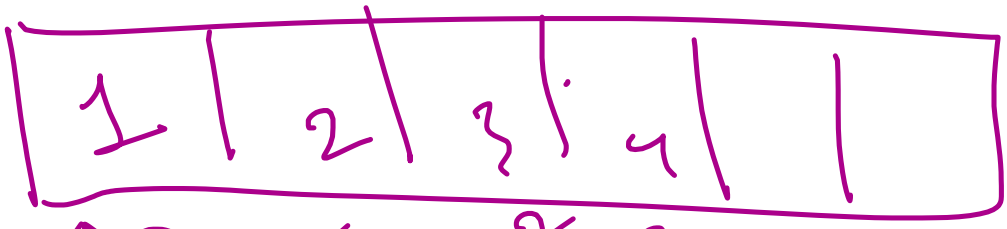
print(st.p

1

Stac

{

arr.



0 1 2 3 sig

top = -1

indexing

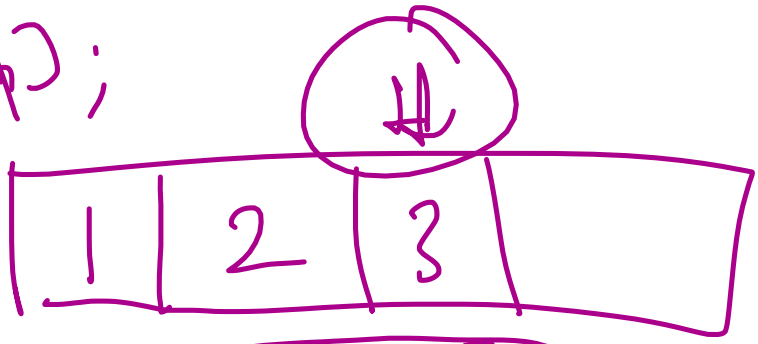
top--

peek.

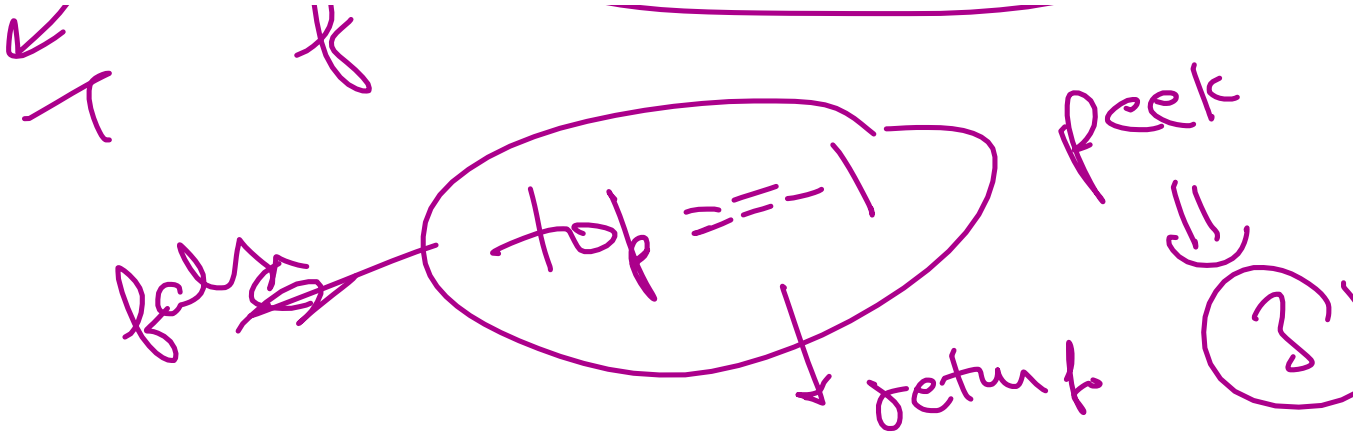
arr[top]

return arr[top];

isEmpty



arr[top] = 3


top == -1
peek
3
return