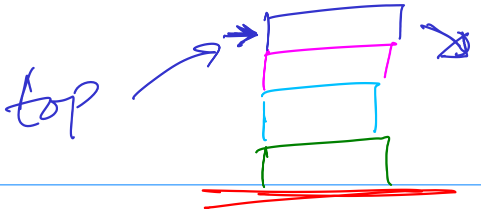


Stack

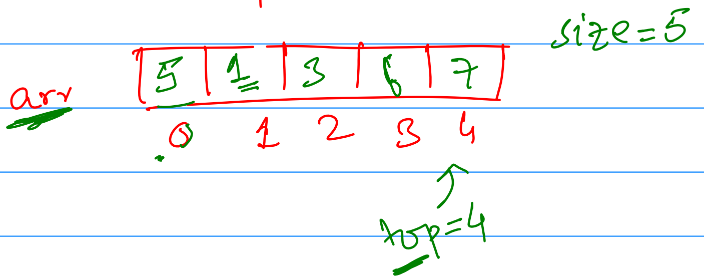
push → insertion at one end

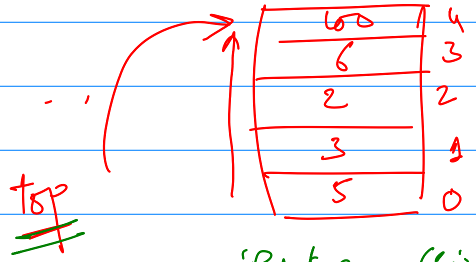
pop → deletion at the same end

Last In First out. → LIFO

→ Full → push : Stack overflow.

→ Empty → pop : Stack underflow size=5





```
class Stack
{ private:
        int top; ✓
        int size; ✓
        int * arr; ✓

  public:
        Stack(int S)
        {   size = S;
            arr = new int[size];
            top = -1; ← empty
        }

        boolean IsFull();
        boolean IsEmpty();
        :
        :
```

```
if (top == (size-1))
    return True
else
    return False
```

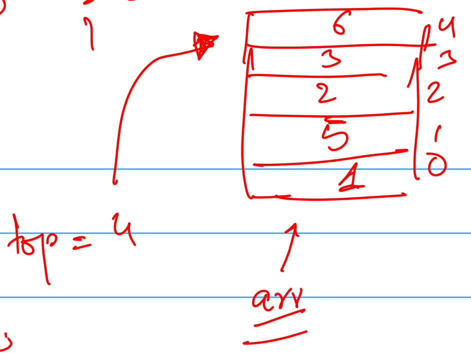
Overflow : IsFull() True False

Underflow : IsEmpty() True False

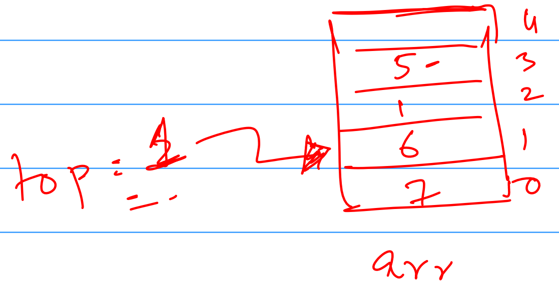
```
if (top == -1)
    return True
else
    return False
```

```
void push(int val);
int pop();
int Top();
```

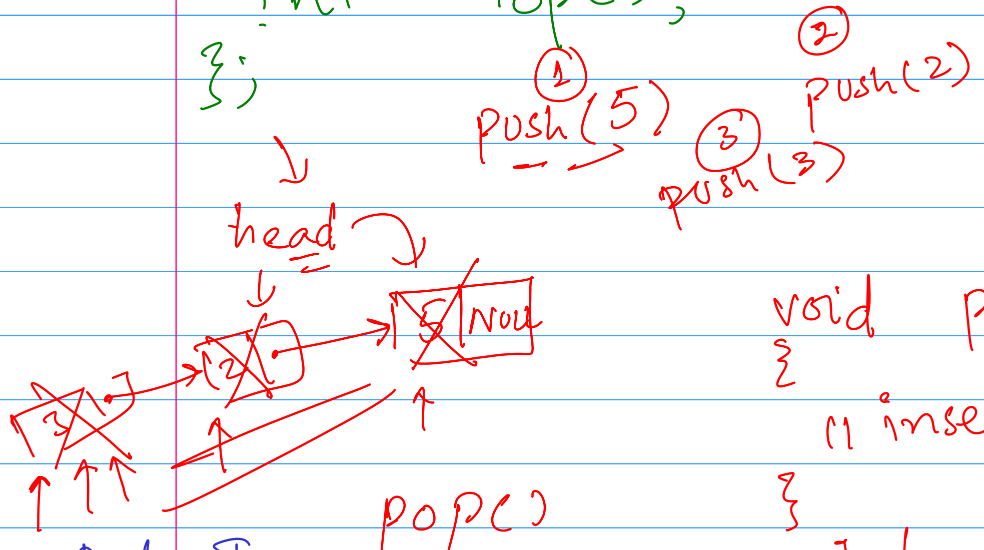
```
Stack s1(5);   size=5
s1.push(1)
s1.push(5)
```



```
void push ( int val )
{
   if ( IsFull( ) )
   {
      Cout << " Stack overflow " << endl;
      Return;
   }
→ arr [++top ] = val ;
}
```



```
int   pop ( )
{
  if ( IsEmpty ( ) )
  {
     Cout<< " Stack Underflow";
     return;
  }
  int val = arr[top];
  → top --;                    }  return arr[top--];
  return val;
}
                 same apart from deletion!
int   Top( );
};
```

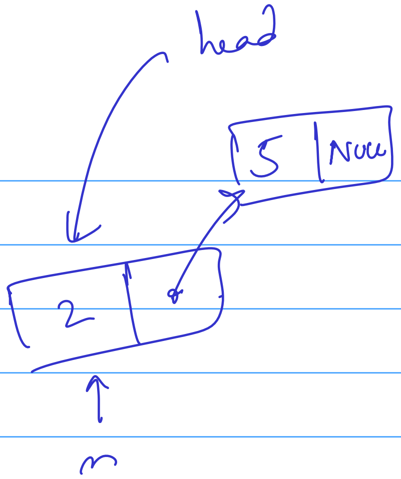


```
node *n=new node(val)
insert(5, 1)

void  push (int  val)
{
   // insert at head.
}

int  Top ()
{ return head->data; }

int  pop ( )
{
 → int  val = head->data;
 → deletion code here
   return val;
}
```



```
node *n = new node(val)
```