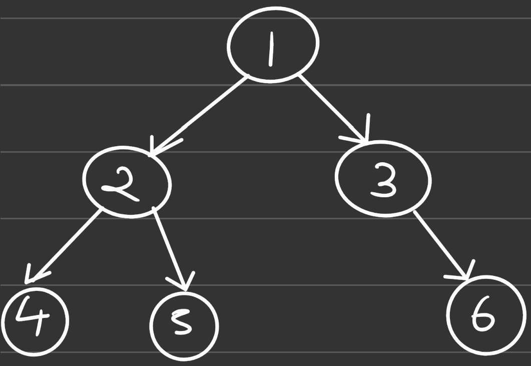
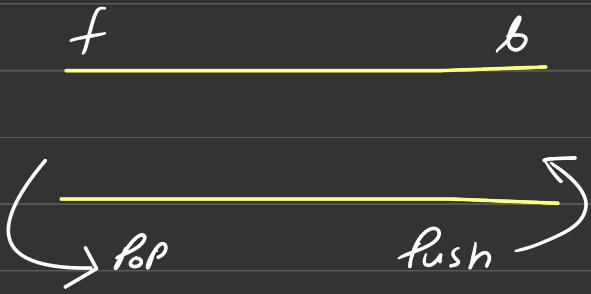
Level Order Traversal (using queue)

PRIYANSH

Level order traversal (using Queue).

(BFS)





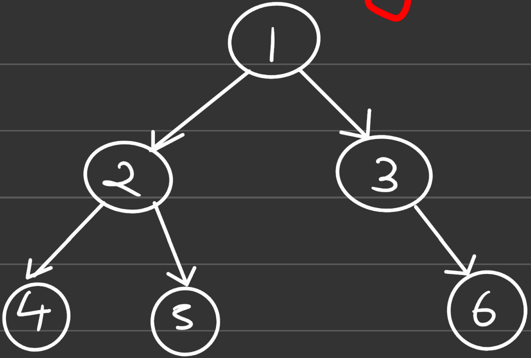
queue<Node*> q;

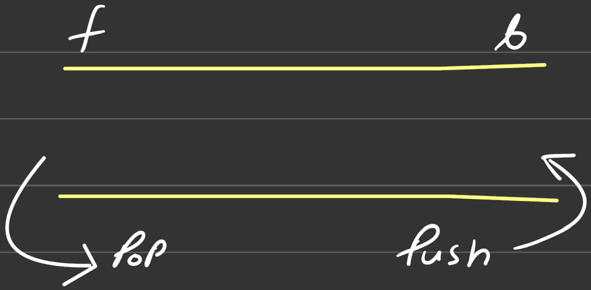
Steps:-

1) Node* temp=q.front(), q.Pop(), Print...

2) push temp->left & temp->right to queue

Working Mechanism:-





POPPED Quantity: 1

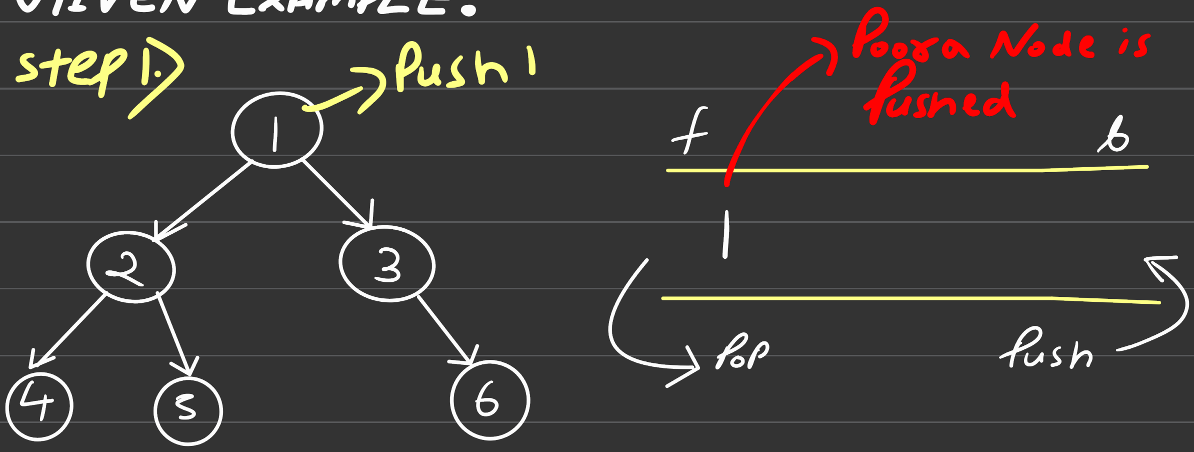
```
Node* temp= q.front();
      q.Pop( );
      cout<<temp->val;
   if(temp->right != Null){
          q.Push (temp->right)
          }
  if (temp->left != NULL){
           q.Push (temp->left)
           }
```

Now checking for the working mechanism of this logic lets just Apply the Code... for the given ABOVE EXAMPLE...

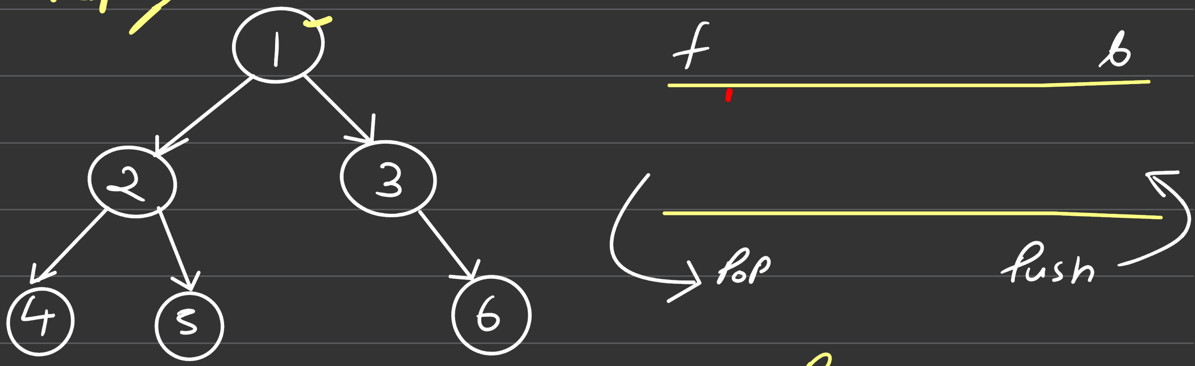
Workflow of Code:-

GIVEN EXAMPLE:

step 1.)



step2)

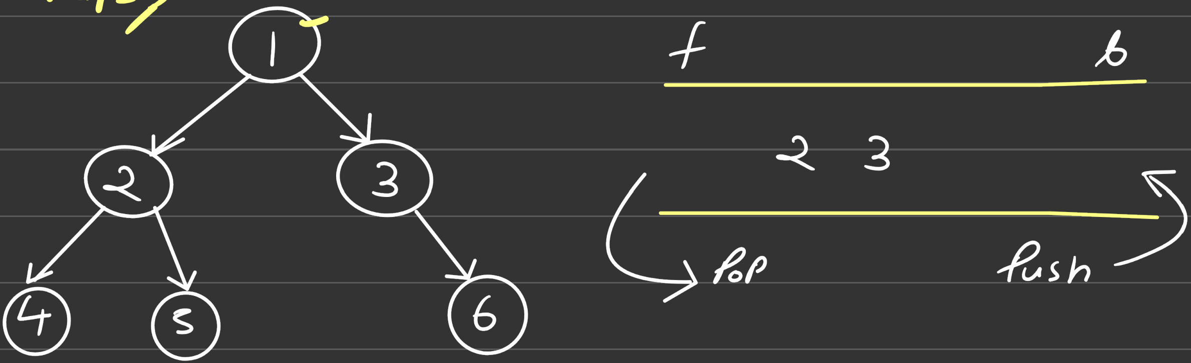


Node* temp= q.top(); Pop;

temp = 1

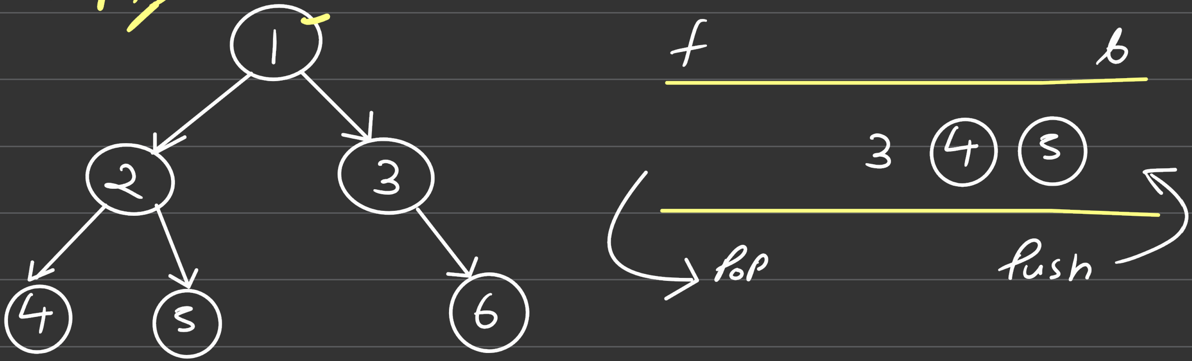
print $\boxed{1}$

step3)



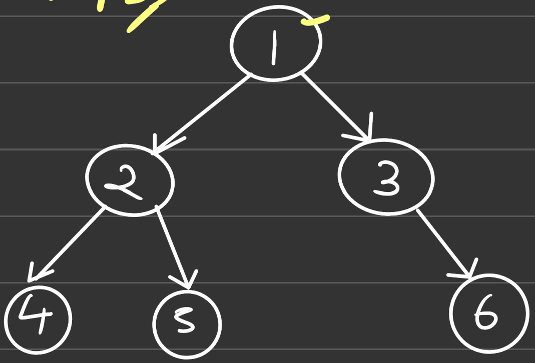
Now we would push the left and right Subtree's root to the Queue

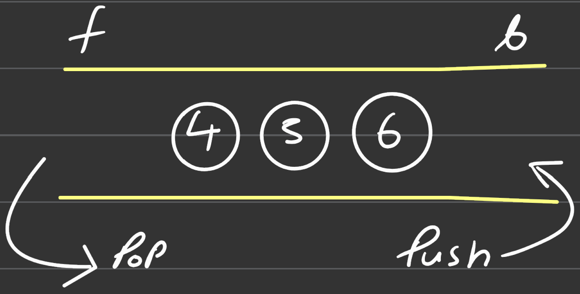
step4)



temp= 2 -> And cout 2

Now push left and right Subtree node of 2 that is 4 and 5 steps)





temp= 3 -> And cout 3

Now push left and right Subtree node of 3 that is 4 and 5

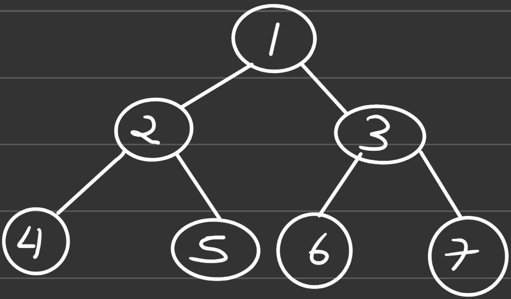
And Now when we have the last level element within the Queue without Any problem and that too in an organised manner...

So Now:- 4 will be printed and popped
then 5 will be printed and popped
6 will be printed and popped

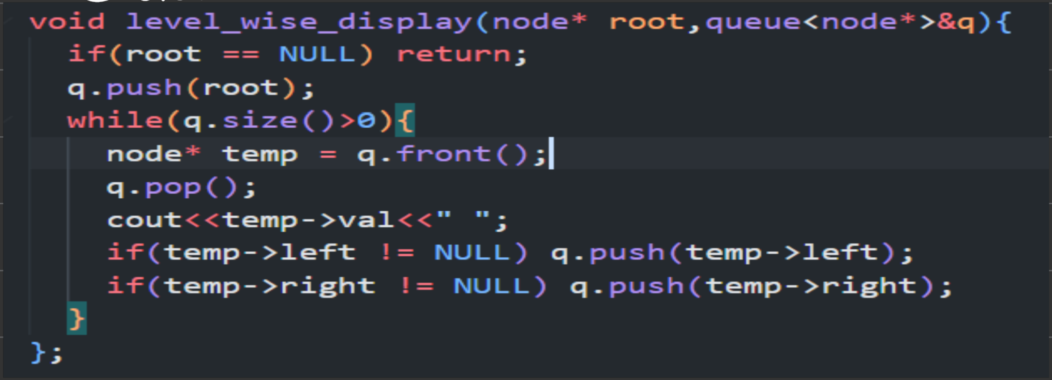
And 4,5,6 me se kisi Bhi node ka left and right Subtree root node push nahi hoga to the Queue as wo Null hoga...

Code Snippet:-

for test case:-

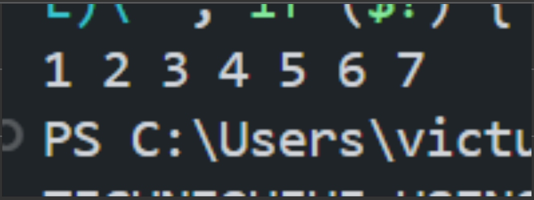


Code:-



```
void level_wise_display(node* root,queue<node*>&q){
    if(root == NULL) return;
    q.push(root);
    while(q.size()>0){
        node* temp = q.front();
        q.pop();
        cout<<temp->val<<" ";
        if(temp->left != NULL) q.push(temp->left);
        if(temp->right != NULL) q.push(temp->right);
    }
};
```

Output:-



```
1 2 3 4 5 6 7
PS C:\Users\victu
```

↳ As same as our Expected output...