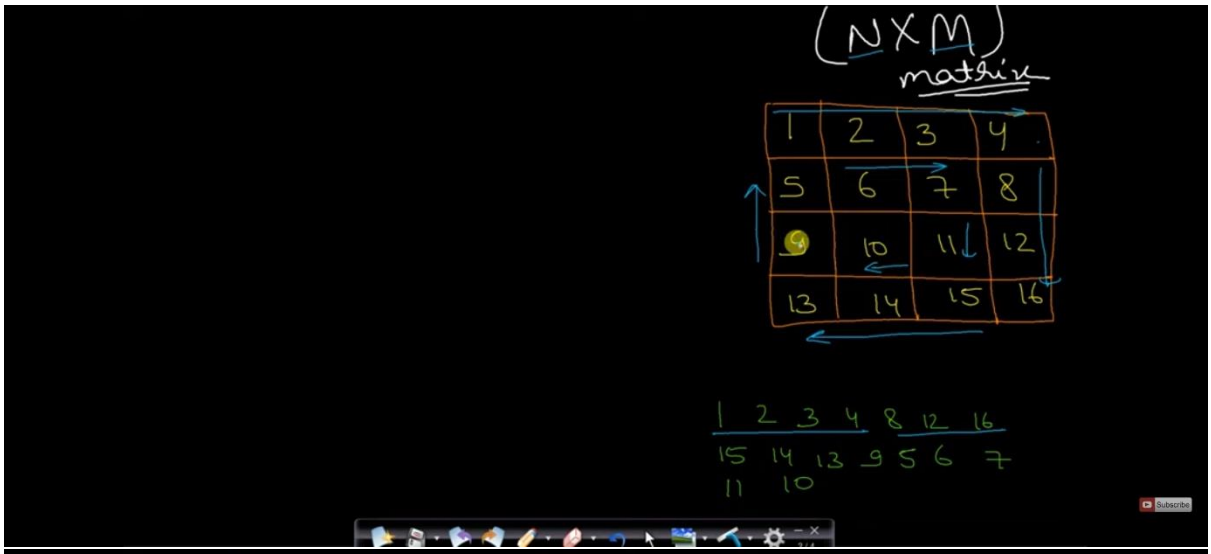
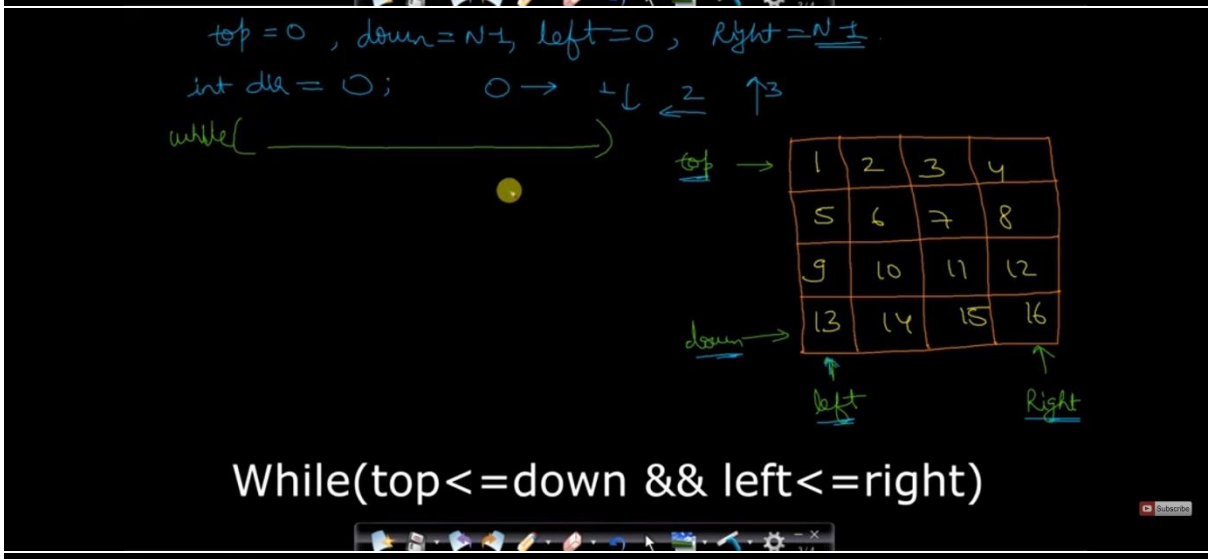
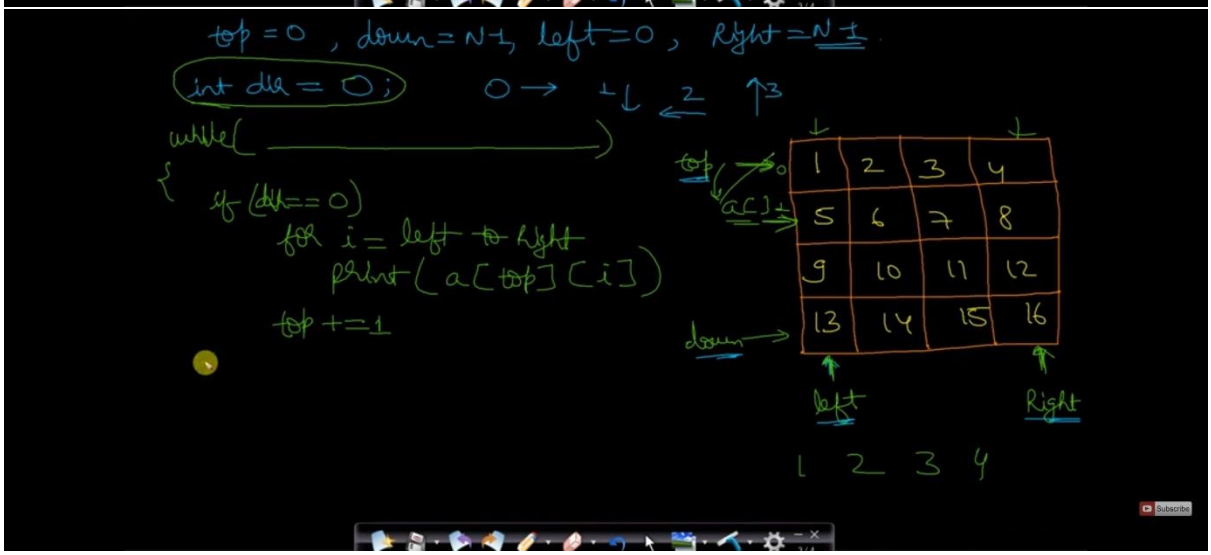




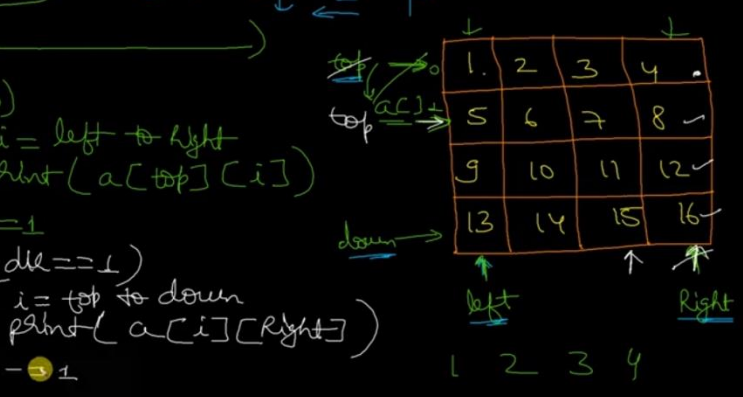
While(top<=down && left<=right)



top = 0 , down = N-1, left = 0 , Right = N-1

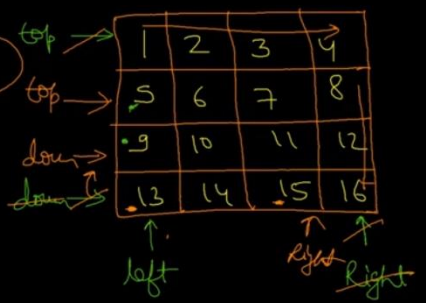
int dir = 0; 0 → 1 ↓ 2 ← 3 ↑

```
while( ____________ )
{
  if (dir == 0)
     for i = left to Right
        print( a[top][i] )
     top += 1
  else if (dir == 1)
     for i = top to down
        print( a[i][Right] )
     Right -= 1
```

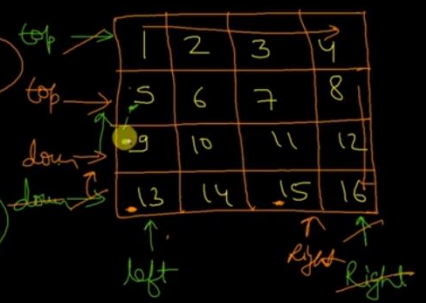


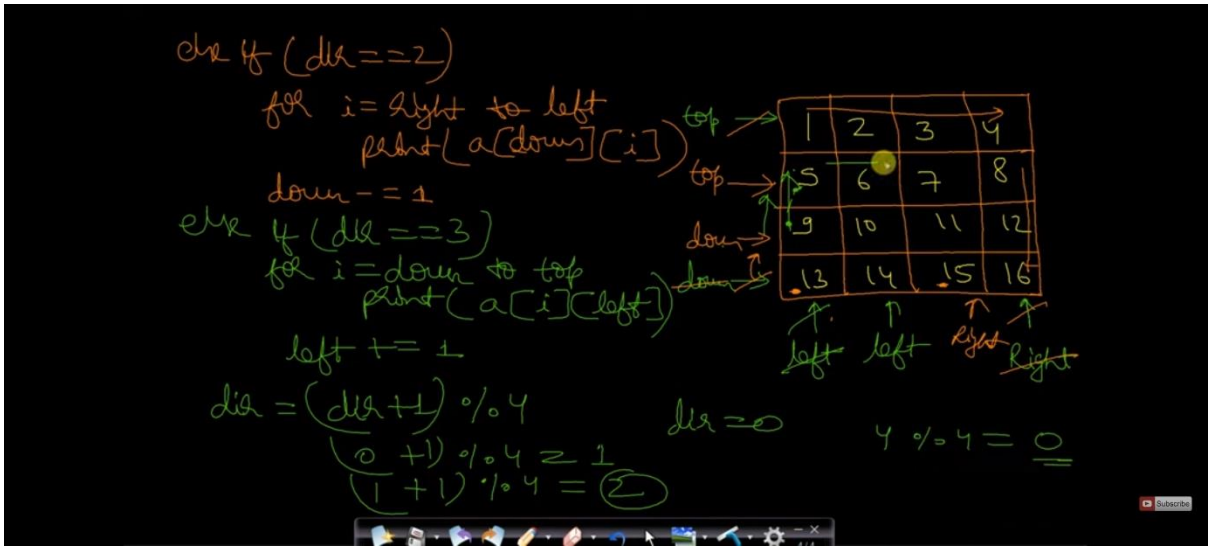
1 2 3 4

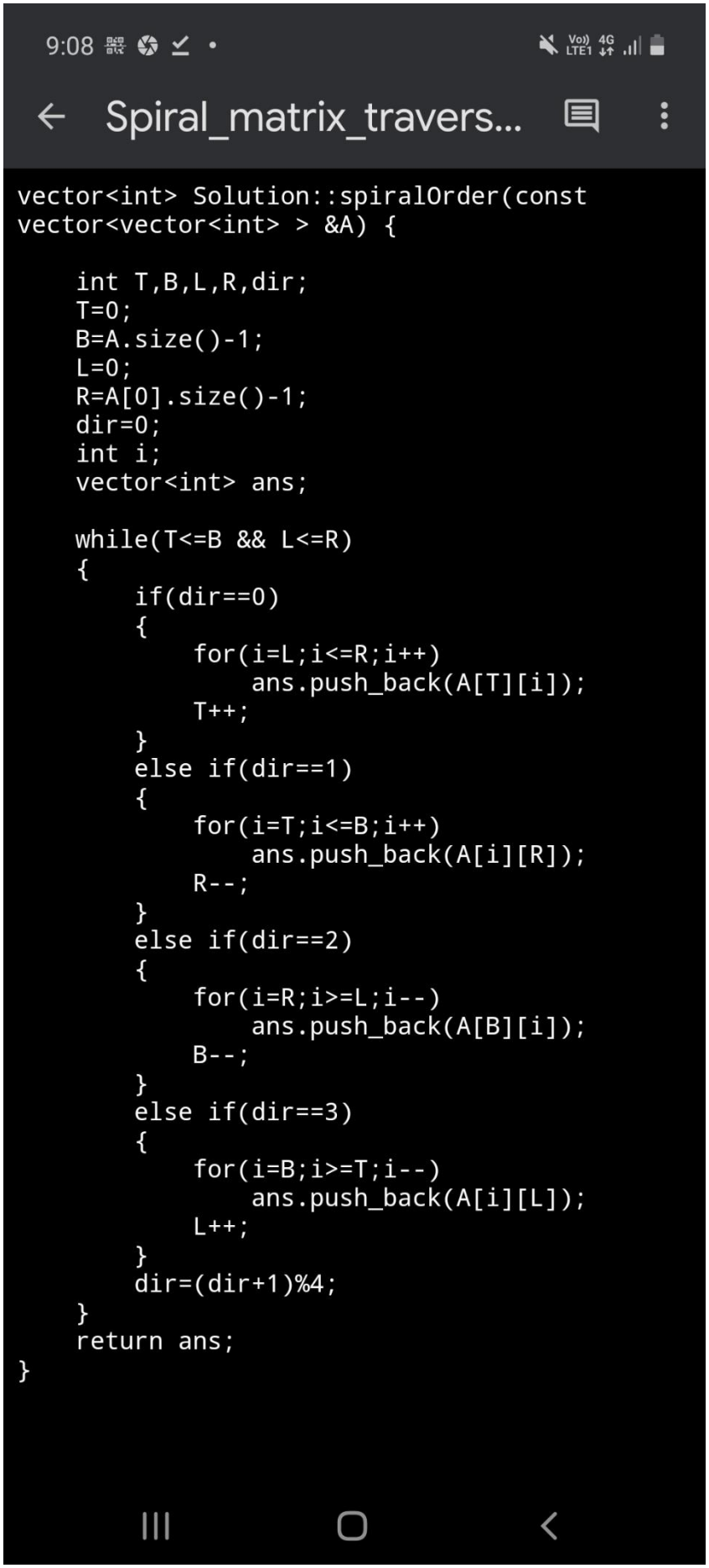
```
else if (dir == 2)
     for i = Right to left
        print( a[down][i] )
     down -= 1
else if (dir == 3)
     for
```




```
else if (dir == 2)
     for i = Right to left
        print( a[down][i] )
     down -= 1
else if (dir == 3)
     for i = down to top
        print( a[i][left] )
```




else if (dir == 2)
for i = right to left
print(a[down][i])
down -= 1
else if (dir == 3)
for i = down to top
print(a[i][left])
left += 1
dir = (dir+1) % 4
(0 +1) % 4 = 1
(1 +1) % 4 = 2
dir = 0
4 % 4 = 0
top
top
down
down
left
left
right
Right
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Subscribe


9:08
Spiral_matrix_travers...
vector<int> Solution::spiralOrder(const
vector<vector<int> > &A) {
int T,B,L,R,dir;
T=0;
B=A.size()-1;
L=0;
R=A[0].size()-1;
dir=0;
int i;
vector<int> ans;
while(T<=B && L<=R)
{
if(dir==0)
{
for(i=L;i<=R;i++)
ans.push_back(A[T][i]);
T++;
}
else if(dir==1)
{
for(i=T;i<=B;i++)
ans.push_back(A[i][R]);
R--;
}
else if(dir==2)
{
for(i=R;i>=L;i--)
ans.push_back(A[B][i]);
B--;
}
else if(dir==3)
{
for(i=B;i>=T;i--)
ans.push_back(A[i][L]);
L++;
}
dir=(dir+1)%4;
}
return ans;
}