Rotation check in Matrix

```
for(int i=1; i<n; i++){
reverse(arr1d, —  —)
reverse ( )
reverse ( )
for( int j=0; i<n ; j++)
arr[i][j] != arr1d[j]
return "NO"
```

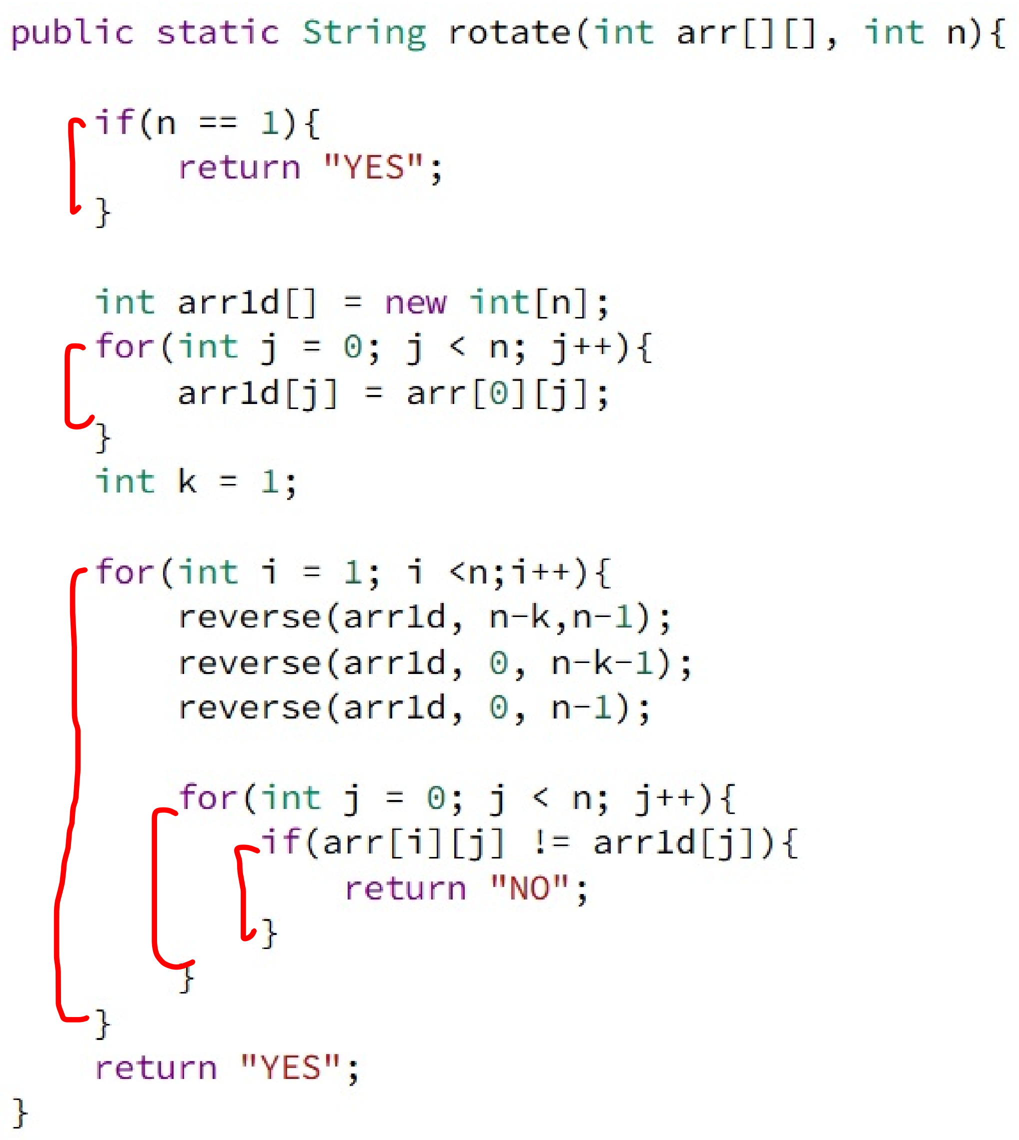
1 2 3

[3 1 2]

1 2 3

3 2 1

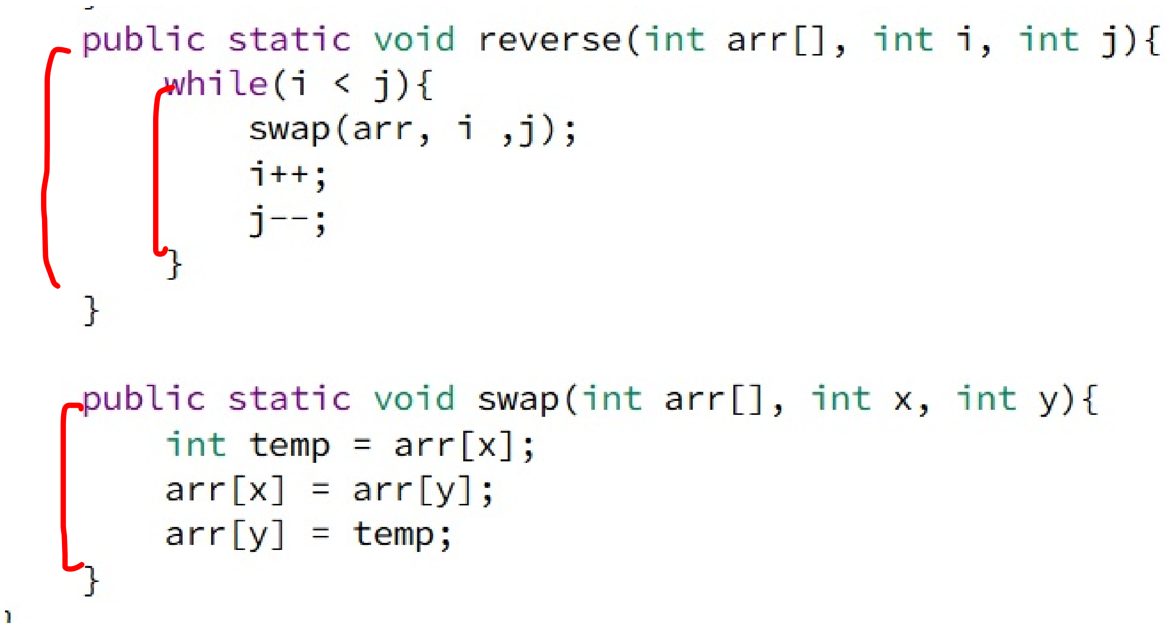
```
int arr1d= new int[n]
for( int j=0; j<n; j++)
arr1d [j] = arr[0][j]
```



```java
public static String rotate(int arr[][], int n){

    if(n == 1){
        return "YES";
    }

    int arr1d[] = new int[n];
    for(int j = 0; j < n; j++){
        arr1d[j] = arr[0][j];
    }
    int k = 1;

    for(int i = 1; i <n;i++){
        reverse(arr1d, n-k,n-1);
        reverse(arr1d, 0, n-k-1);
        reverse(arr1d, 0, n-1);

        for(int j = 0; j < n; j++){
            if(arr[i][j] != arr1d[j]){
                return "NO";
            }
        }
    }
    return "YES";
}
```

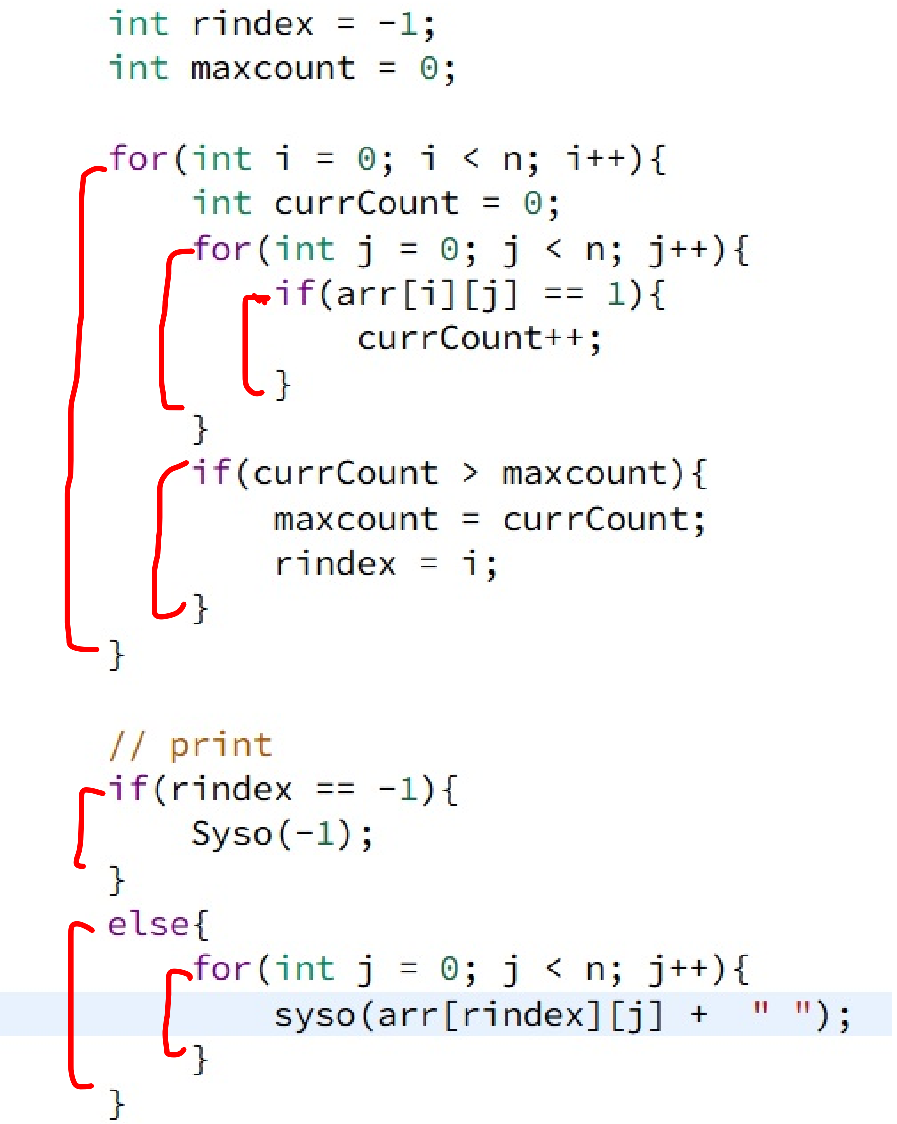
SC → O(n)

TC → O(nxn)



```java
public static void reverse(int arr[], int i, int j){
    while(i < j){
        swap(arr, i ,j);
        i++;
        j--;
    }
}

public static void swap(int arr[], int x, int y){
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```

Count max 1's in matrix

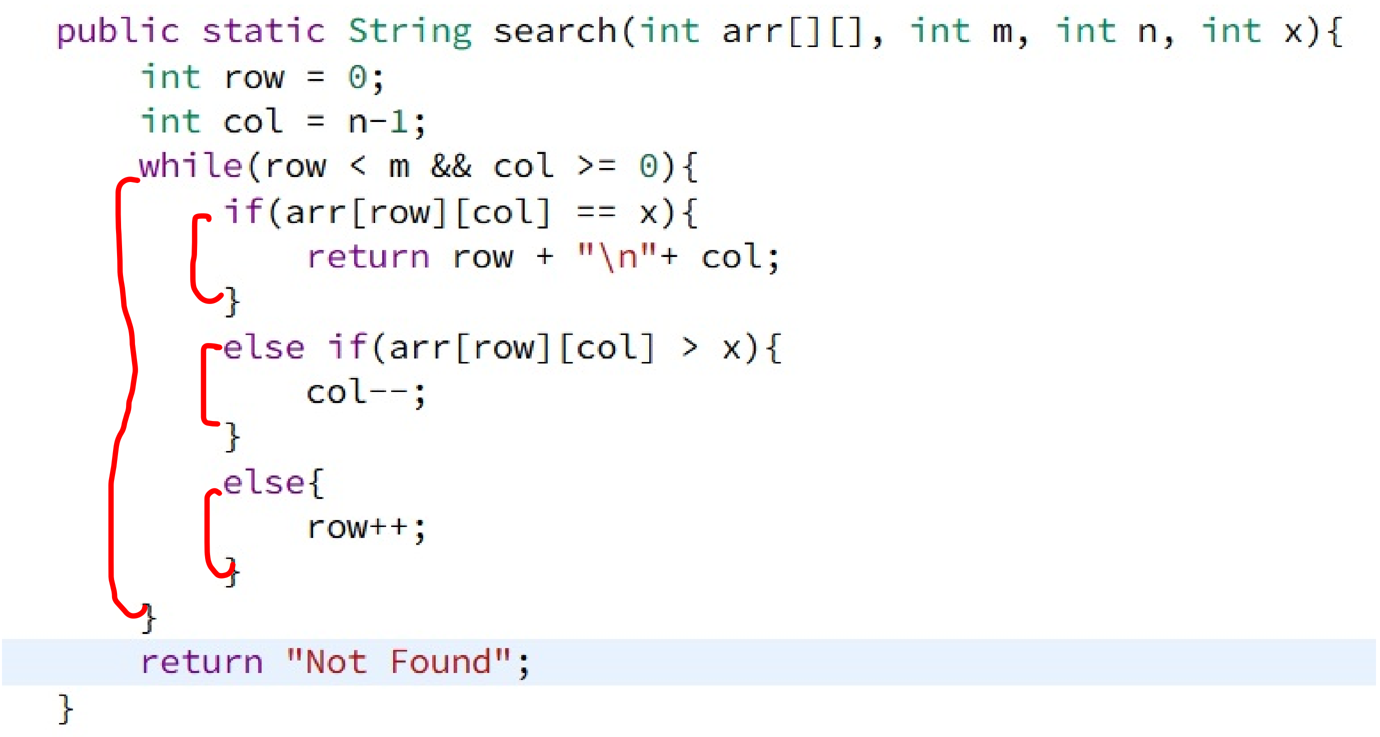


```
int rindex = -1;
int maxcount = 0;

for(int i = 0; i < n; i++){
    int currCount = 0;
    for(int j = 0; j < n; j++){
        if(arr[i][j] == 1){
            currCount++;
        }
    }
    if(currCount > maxcount){
        maxcount = currCount;
        rindex = i;
    }
}

// print
if(rindex == -1){
    Syso(-1);
}
else{
    for(int j = 0; j < n; j++){
        syso(arr[rindex][j] +  " ");
    }
}
```

TC → O(n×n)

SC → O(1)

HW_Search in a sorted matrix



```
public static String search(int arr[][], int m, int n, int x){
    int row = 0;
    int col = n-1;
    while(row < m && col >= 0){
        if(arr[row][col] == x){
            return row + "\n"+ col;
        }
        else if(arr[row][col] > x){
            col--;
        }
        else{
            row++;
        }
    }
    return "Not Found";
}
```

TC → O(n)

SC → O(1)