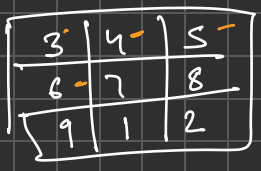
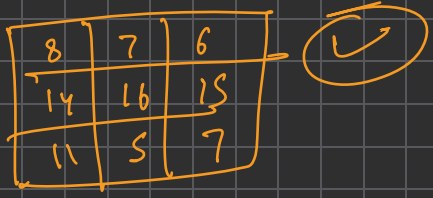
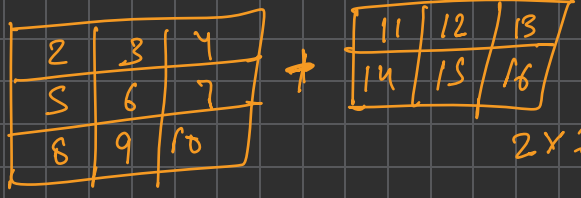
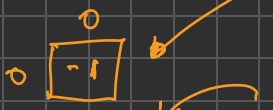




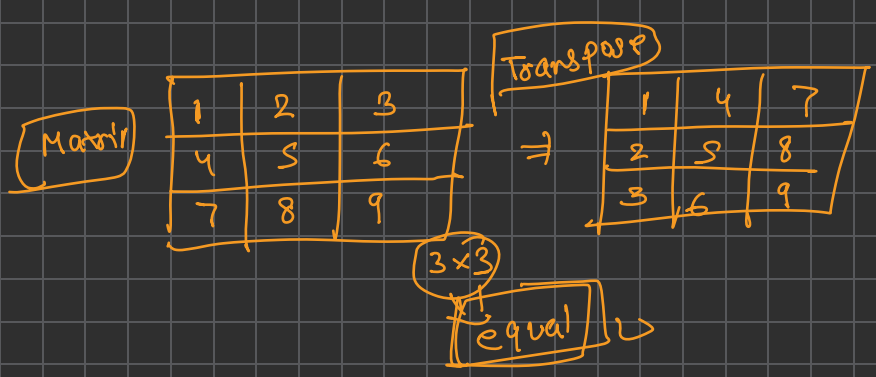


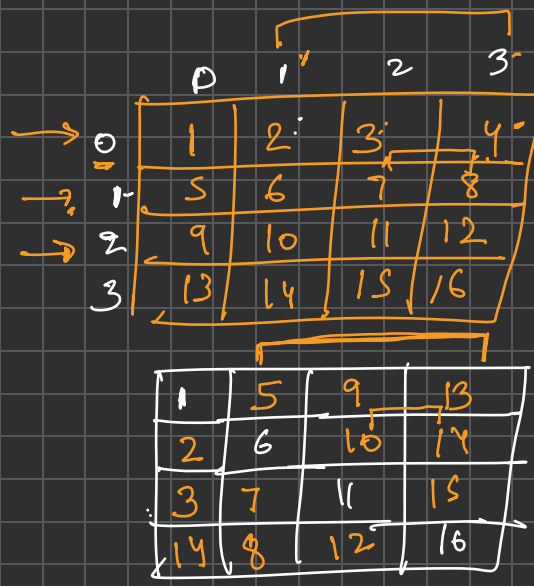
add ho sakta ?

vector<vector<int>> ans(1, vector<int>(1))

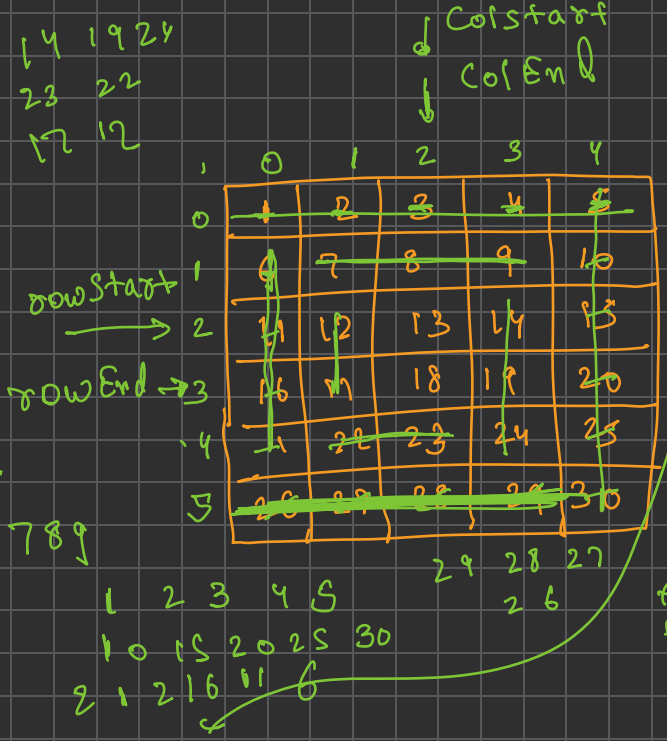


ans[0][0] = -1






```
for(i=0; i<rowsize-1; i++)
{
for(j=i+1; j<colsize; j++)
{
swap(mat[i][j], mat[j][i]);
}
```



14 1924
23 22
17 12

789

1 2 3 4 5
10 15 20 25 30
21 26 11 6

29 28 27
26

① Row no print karo rowstart tk

② Print Last column except the first elem

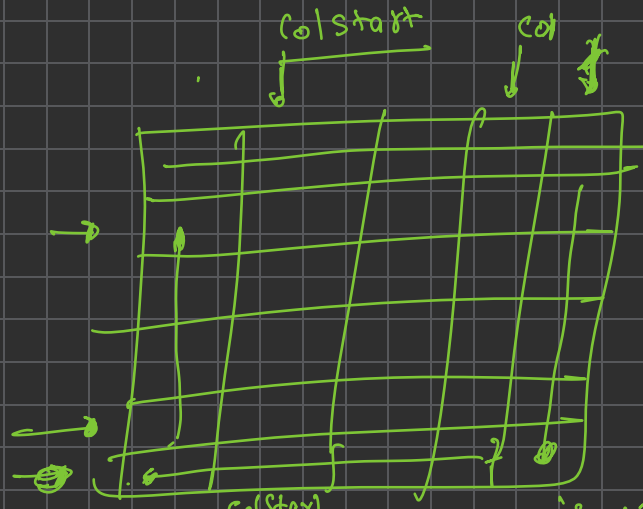
③ Print the last row in reverse order except 1st

④ Print the first column in reverse order except first & last elem

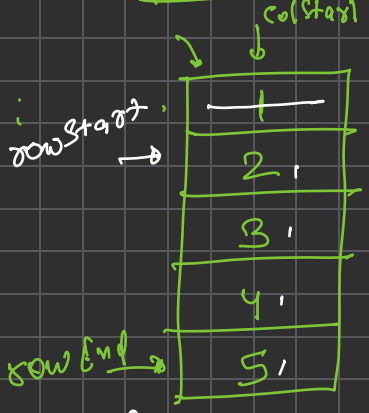
```
for(int i = rowStart; i<=rowEnd;i++)
    cout << mat[i][colEnd];

for(int j = colEnd; j>=colStart;
        j--)
    cout << mat[rowEnd][j];
```

```
For(int j = colstart;
        j<=colEnd;j++)
    cout << mat[rowStart
        [j];
```

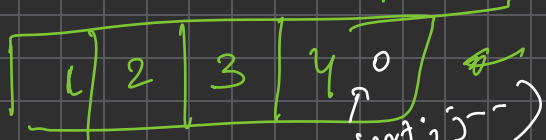
```
for(int i=rowEnd; i>=rowStart; i--)
{
    cout << mat[i][colstart];
}
```



Print
1. first row
2. Last col
3. Last row
4. first col



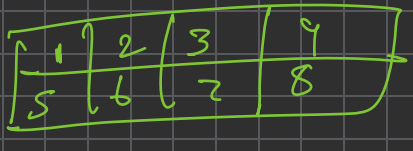
is wale logic Legay ustrp apply

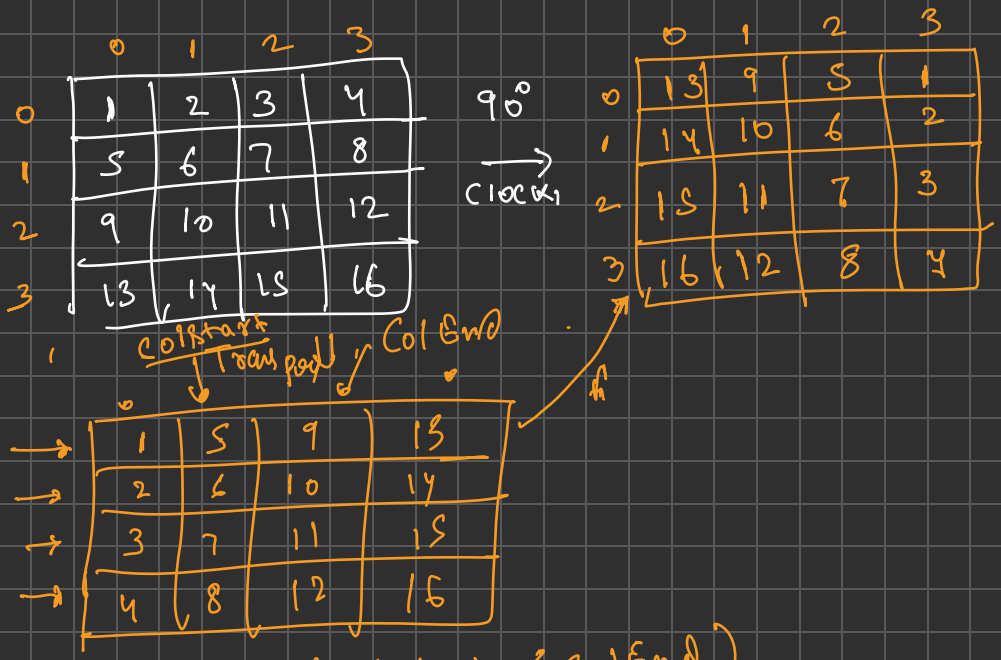


1 2 3 4 5

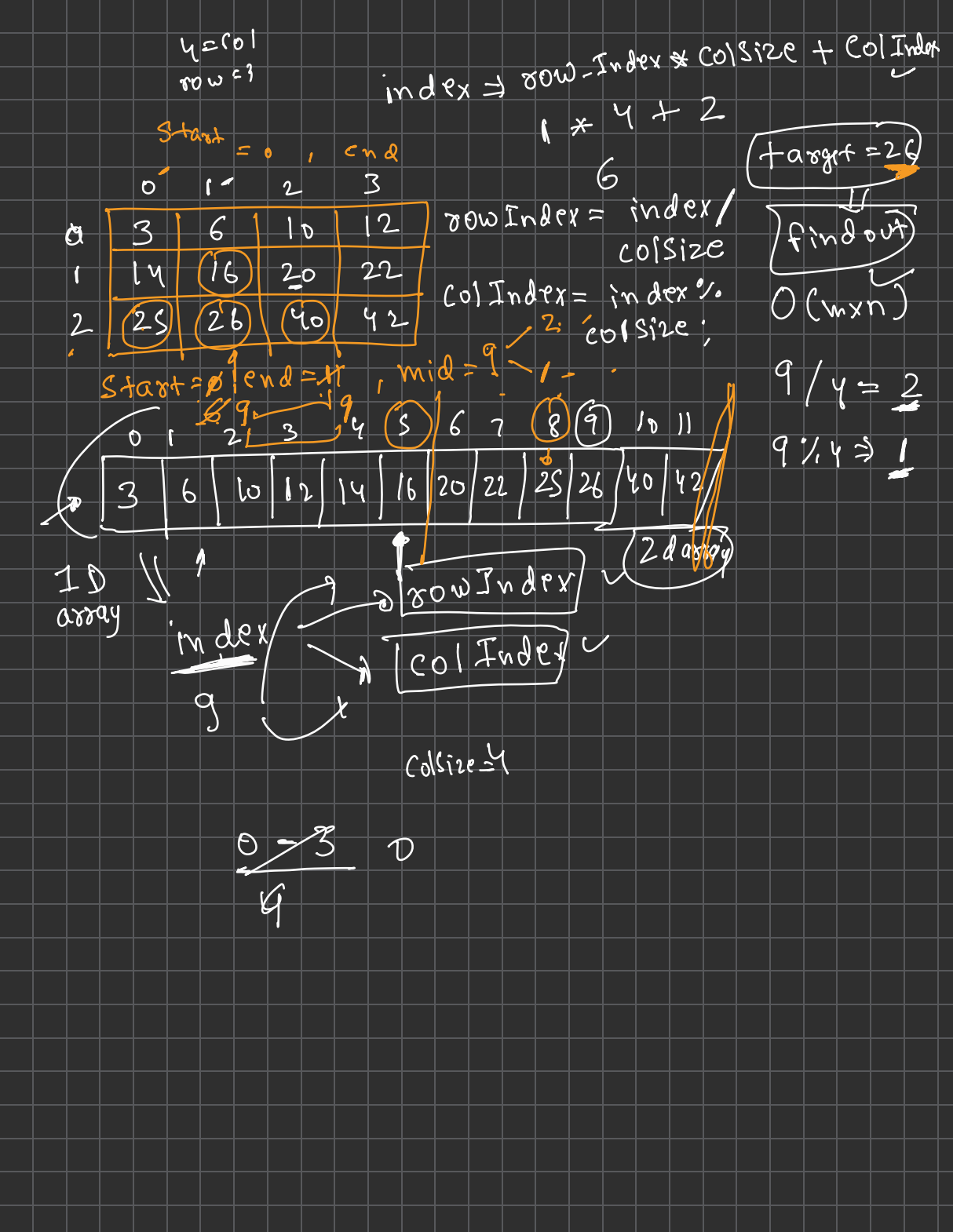
```
for(int j=colEd; j>=colstart; j--)
```

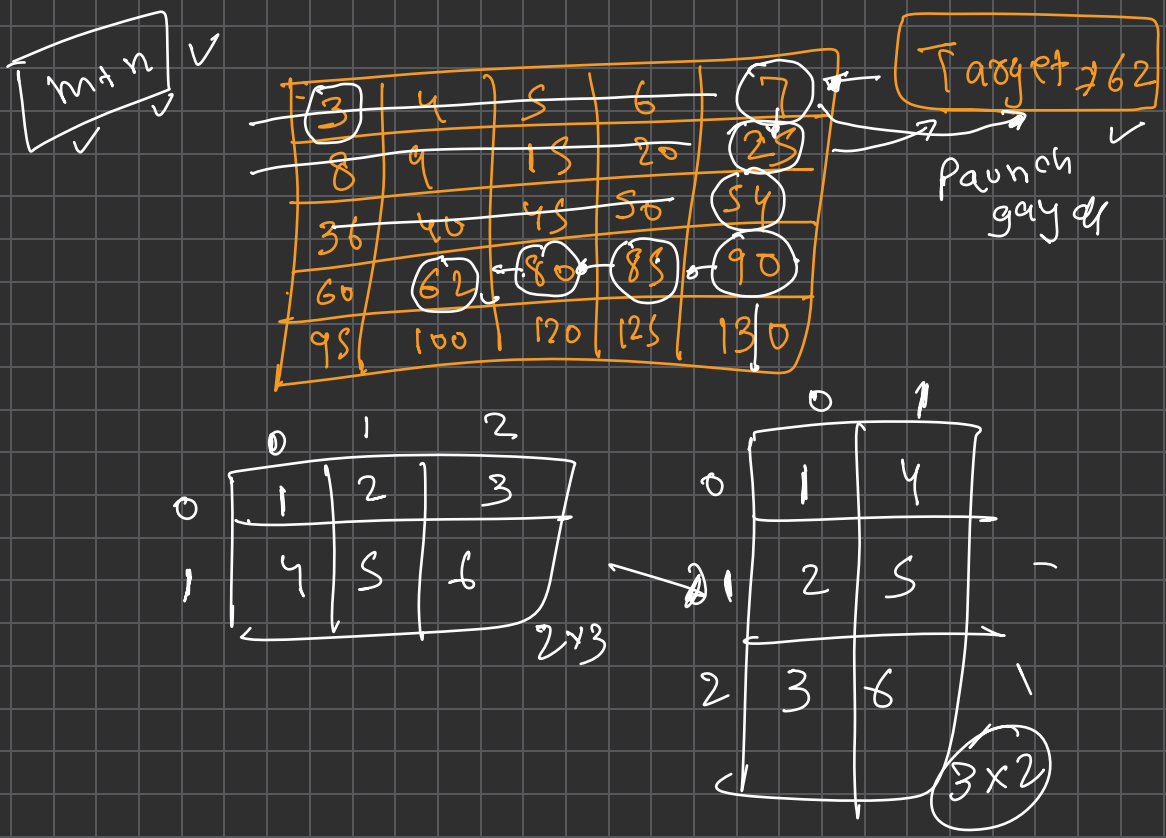
-1






```
while (colstart < colEnd)
{
  for (i=0; i<rowSize; i++)
    swap (mat[i][colstart], mat
          [i][colEnd])
}
```


4=col
row=3
index ⇒ row-Index * colSize + colIndex
1 * 4 + 2
6
start = 0 , end
0 1 2 3
0 3 6 10 12
1 14 16 20 22
2 25 26 40 42
rowIndex = index / colSize
colIndex = index % colSize;
target = 26
findout
O(mxn)
9/4 = 2
9%4 ⇒ 1
start = 0 , end = 11 , mid = 9
0 1 2 3 4 5 6 7 8 9 10 11
3 6 10 12 14 16 20 22 25 26 40 42
2d array
1D array
index
9
rowIndex
colIndex
colSize = 4
0 3
9
0


min
3 4 5 6 7
8 9 15 20 25
36 40 45 50 54
60 62 80 85 90
95 100 120 125 130
Target = 62
Paunch gayi
0 1 2
0 1 2 3
1 4 5 6
2×3
0 1
0 1 4
1 2 5
2 3 6
3×2