DAY-2.

ROTATE IMAGE

Take transpose of Matrix and reverse.

↳
```
for (i=0; i<n; i++)
    for (j=i+1; j<n; j++)
        swap (matrix[j][i], matrix[i][j])
```

Now Rotate Reverse

```
for (i=0; i< n; i++)
    for(j=0; j< n/2; j++)
        swap ( matrix[i][j], matrix[i][n-j-1])
```

```
int cols=0, colend=m-1   while ( cols < cole){
    for(int row=0, row < matrix[0].size(); row++) {
        swap( matrix[ row ][colstart], matrix[row][colend]);
    }
    cols++
    cole--; }
```

SPIRAL TRAVERSAL OF MATRIX

startingrow=0, startingcol=0, endingrow=r-1, endingcol=c-1; count=0; total=r*c

while(count<total)

1. (sr; count<total && i<=endingcol)
   matrix[sr][i]; sr++;

2. (sr; i<=endingrow)
   matrix[i][endingcol], endingcol--

3. (ec; i>=sc, i--)
   matrix[er][i]; cr--;

4. (er; i>=sr, i--)
   matrix[i][sc]; sc++;

SEARCH AN ELEMENT IN MATRIX.

Use Binary search

matrix[mid/col][mid%col] to get the index if < s=m+1 else e=m-1

FIND MEDIAN IN A ROW-WISE SORTED

Push all element in ans, then sort, then get ans[ans.size()/2]

ROW WITH MAXIMUM NO. of 1s.

```
count=0; max=0; ans=0;
    for(i<n i++){
        for (j<m j++){
            if (found 1) count++
    if (max<count) then if (max==0) ans=-1
        max=count                  else return ans,
        ans=i
```

SORT ALL ELEMENTS IN MATRIX Push all in Vector then sort then create matrix ans, Pointer and matrix[i][j] = vector[count] p++; /

JS Day-6.

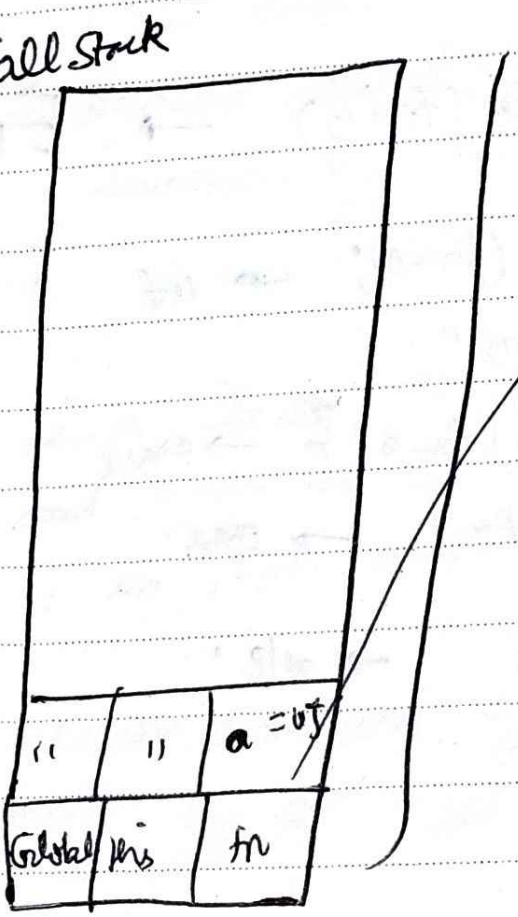
```
function fn(){
    console.log(a);
    var a=10;
    console.log(a);
    if(a==10){
        var a;
        console.log(a);
    }
    console.log(a);
}
fn()
```

uf

10

10

10



मई 2019

17 शुक्रवार

2019 अप्रैल

रवि	सोम	मंगल	बुध	गुरु	शुक्र	शनि
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

```
Var  a =10;
console. log (a);                    → 10
function fn() {
  console. log (a);                  → undefined
  Var a =20;
   a ++;
   console. log(a)                   → 21
if(a ) {
     Var a =30;
      a++
     console. log (a);               → 31
  }
  Console. log(a)                    → 31
  }
fn()
console. log(a);                     → 10
```

Function scope. Each fun creates new scope. Variables defined inside a fun are not accessible from outside the fun.

it is call fun scope bcoz using var keyword it will be accessible throughout the function.

18 शनिवार

```
Ex →   let fruits = "apple";
       console. log (fruits);        → apple
       {
if     console. log (fruits)         → ERROR [TDZ]
       let fruits;
       console. log (fruits);        → uf
       fruits = "orange";
       { console. log (fruits) }     → orage
       Console - log fruits          → orage.
       }
console. log (fruits                 → apple.
```

BLOCK SCOPE:
Variables declared inside a {} block cannot be accessed from outside the block.

	Scope	Reassign	Redeclare	Temporal Dead zone
Var	Function	✓	✓	✗
let	block	✓	✗	✓
const	block	✗	✗	✓

TEMPORAL DEAD ZONE is the area of block where a variable is inaccessible until the moment the computer initializes it with a value.

A blocks temporal end zone start at beginning of block's local scope. It ends when computer fully initializes variable with a value.

When computer hoists a var variable, it automatically initializes the variable value undefined.

In contrast JS does not initialize let (or const) with any value whenever it hoists.

Therefore, let variable's TDZ ends when JS fully initializes it with value specified during its declaration.

However, var TDZ end immediately after hoisting.

SHADOWING →
when a variable is declared in a scope having the same name defined on its outer scope and when we call variable from inner scope value assigned in inner scope is value stored in variable.

ILLEGAL → It should not cross boundary of a scope we can shadow a var variable by let but cannot do opposite.