Page No.:
Date: / /

Lecture - 17

2D - Array

* Two Dimensional Array

row ↓

Name : [Rohit, Paridhi, Riya, Jack, John]
M_NO : [92364, 9432, 9192, 9061, 9054]
R_NO : [32, 64, 66, 61, 39]

→ Column

1	3	5	8
2	4	6	7
9	12	13	15
16	14	14	16

2D Matrix

Initialise :

```
int arr[row][column]
int arr[3][4]
```

3 row & 4 column.

Array : A[3][4]

	0	1	2	3
0	00	01	02	03
1	10	11	12	13
2	20	21	22	23

3x4

Teacher's Signature .....

How it stored in Memory?

Row Major

0	1	2	3	4	5	6	7	8	9	10	11
00	01	02	03	10	11	12	13	20	21	22	23

How to find which element are present in which row major?

Index = (row_index) * column + column_index;

Ex: for 20

2 x 4 + 0 = 8

At 8th index 20 is present.

Index of Row Major is given:

Find Row index & Column index.

Row_index : Index / column

Column_index : Index % column.

* col-Major order:

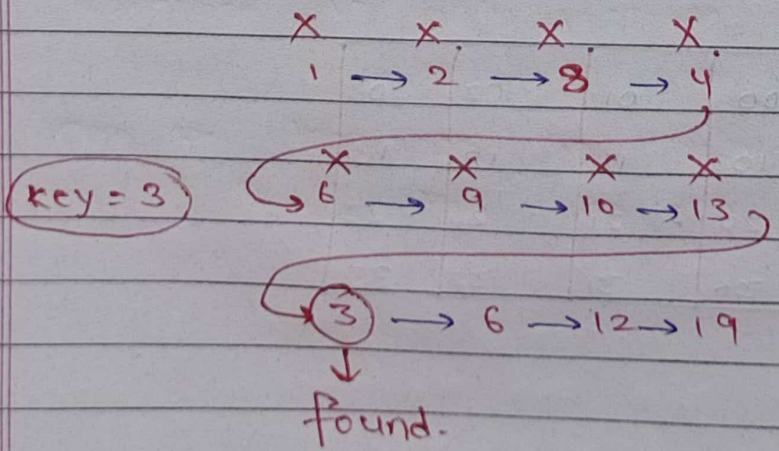
00	01	02	03
10	11	12	13
20	21	22	23
30	31	32	33

Teacher's Signature.....

```
int main(){
    int arr[3][4];
  for(int i=0; i<3; i++){
      for(int j=0; j<4; j++){
          cin >> arr[i][j];
      }
  }
  for(int i=0; i<3; i++){
      for(int j=0; j<4; j++){
          cout << arr[i][j];
      }
      cout<<endl;
  }
```

3×4	0	1	2	3
0	00	01	02	03
1	10	11	12	13
2	20	21	22	23

* Searching an element:



Code:

```
int arr[3][4];
for(int i=0; i<3; i++){
    for(int j=0; j<4; j++){
        cin>> arr[i][j];
    }
}
int key =19;
for(int i=0; i<3; i++){
    for(int j=0; j<4; j++){
        if(arr[i][j] == key){
            return true;
        }
    }
}
return false;
```

O(M x n)

* Sum of all element of Array:

1 + 2 + 3 + 4 = 10

1	2	3	4	
5	6	7	8	10+5+6+7+8 = 36
9	10	11	10	36+9+10+11+10 = 76

Code:

```
int arr[3][4];
for(int i=0; i<3; i++){
    for(int j=0; j<4; j++){
        cin >> arr[i][j];
    }
}
```

Teacher's Signature.....

```
int sum = 0;
for (int i=0; i<3; i++){
    for (int j=0; j<4; j++){
        sum += arr[i][j];
    }
}
```

* Transpose of a 2D Matrix :

1	2	3	Transpose	1	4	7
4	5	6	⟷	2	5	8
7	8	9		3	6	9

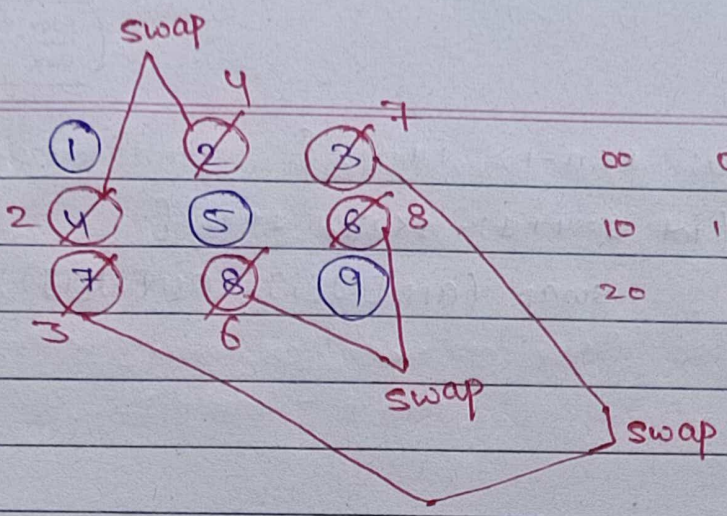
00	01	02		00	10	20
10	11	12	⟶	01	11	21
20	21	22		02	12	22

reverse the index
i to j & j to i
Ans[i][j] = A[j][i]

Transpose is only possible if size is equal.

* Can we transpose the array without creating a new array?

Teacher's Signature



00	01	02
10	11	12
20	21	22

1	4	7
2	5	8
3	6	9

Exchange value

[i][j] With [j][i]

01 With 10

Change Only Half Matrix

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

→ Change Made only one this

Diagonal Does not change

if (i == j)

No change

Teacher's Signature.....

```
for(int i=0; i<n-1; i++){
    for(int j=i+1; j<m; j++){
        swap(arr[i][j], arr[j][i]);
    }
}
```

* Flip the array:

0 - 2 = 2
1 - 1 = 2

1	2	3
4	5	6
7	8	9

↓

7	8	9
4	5	6
1	2	3

Code:

```
for(int i=0; i<n/2; i++){
    for(int j=0; j<m; j++){
        swap(arr[i][j], arr[row-1-i][j]);
    }
}
```

Teacher's Signature.....

Page No.:
Date: / /

* Rotate Image : (90°)

1	2	3		7	4	1
4	5	6	90° →	8	5	2
7	8	9		9	6	3

1	2	3		1	4	7
4	5	6	Transpose →	2	5	8
7	8	9		3	6	9

↓ Flip vertical

7	4	1
8	5	2
9	6	3

* Search in a 2D Matrix (sorted):

1	3	5	7
10	11	16	20
23	30	34	60

Target = 16

sorted : then apply binary search

0	1	2	3	4	5	6	7	8	9	10	11
1	3	5	7	10	11	16	20	23	30	34	60

```
start = 0
end = 11
mid = (strat + end) / 2;
row_index = mid / column;
column_index = mid % column;
```

Teacher's Signature.....

```
if (arr[row_index][col_index] == target) {
        return mid;
    }
else if (arr[row_index][col_index] < target) {
        start = mid + 1;
    }
else {
        end = mid - 1;
    }
```

* 2nd Approach :

1	2	5	7
11	13	18	20
21	27	30	35

key = 18

18 Kis row Me Rahega

18 > 1 && 18 < 11 false

18 lies in 2nd row

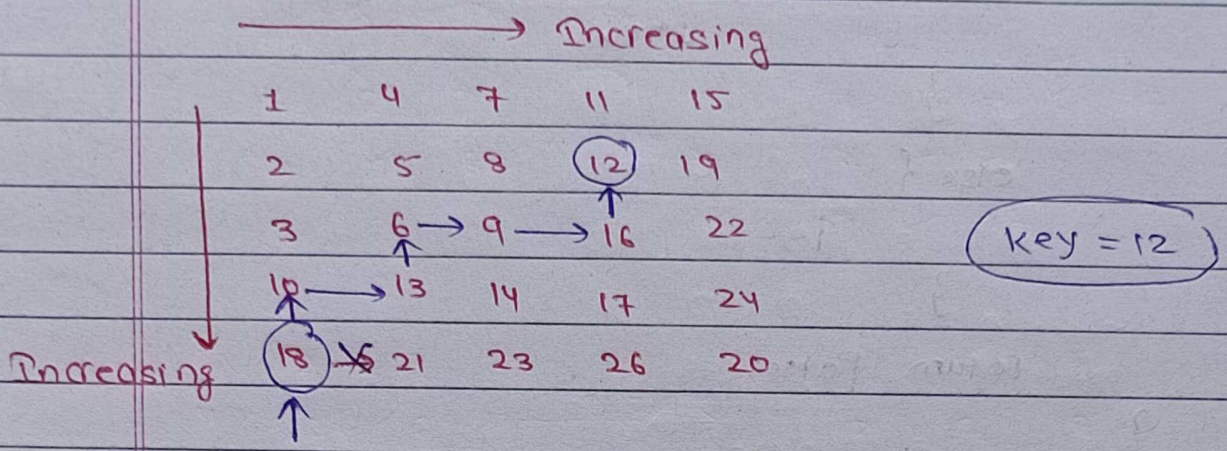
Now Apply binary search in 2nd row :

11 13 (18) 20

↑

key.

* Search in a 2D Matrix:



From this corner

one pattern is found.

If go up ↑ : value decreased

If go right → : value increased

12 < 18 → go up (↑)

10 < 12 → go right (→)

13 > 12 → go up (↑)

6 < 12 → go right (→)

9 < 12 → go right (→)

16 > 12 → go up (↑)

12 = 12 → found.

Code:

```
int row = matrix.length;
int col = matrix[0].length;
int i = row-1, j = 0;
while (i>=0 && j<col){
    if(matrix[i][j] == target)
        return true;
```

Teacher's Signature.....

```
        else if (matrix[i][j] < target){
                j++;
            }
        else {
                i--;
            }
        return false;
    }
```

Teacher's Signature .....