Suppose there are 10 students, we need to store their marks

```
int [] marks = new int [10];
// take input and store values in array
```

3 new admissions

```
int [] newarr = new int [13];
```

3 new input

// copy the first 10 students marks from old array and then take 3 new inputs

ArrayList → change the size [dynamic]

```
ArrayList <Integer> al = new ArrayList <>();
```

(Integer: data type; al: name)

§ Double / Long / Float / String §

```
ArrayList<Integer> al = new ArrayList<>(); → al : []

// add new elements
   // al.add(element);
   al.add(10);
   al.add(1);
   al.add(5);
```

	size
al : [10]	1
al : [10, 1]	2
al : [10, 1, 5] (0 1 2)	3

```
S.O.P(al.size());  ⟶  3

al.add(20);
```

al : [10, 1, 5, 20] (0 1 2 3)	4

```
// get element
   // al.get(index);
   al.get(1);  ⟶  1
   al.get(3);  ⟶  20
   al.get(4);  ⟶  Error
```

// change existing elements

// al.set(index, element);

al.set(2, 50);

al: [10, 1, 50, 20] 4
    0 1 2 3

S.O.P(al); ⟶ [10, 1, 50, 20]

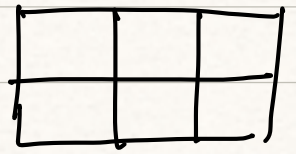
// remove elements

al.remove(2);
   ↳ index

al: [10, 1, 20] 3
    0 1 2

2D - Arraylist

```
int [][] arr = new int [2][3]; // 2D Array
```



2D Arrays → an array of arrays

2D Arraylist → an arraylist of arraylist

```
Arraylist <Integer> l1 = new Arraylist <>();
```

```
l1.add (10);
l1.add (20);
```

l1 : [10]

l1 : [10, 20]

```
Arraylist <Integer> l2 = new Arraylist <>();
```

```
l2.add (-1);
l2.add (5);
l2.add (9);
```

l2 : [-1]

l2 : [-1, 5]

l2 : [-1, 5, 9]

// creating 2D ArrayList

```
ArrayList < ArrayList <Integer> > al = new ArrayList <>();
```

```
al.add (l1);
al.add (l2);
```

al : [[10, 20]]

al: [[10, 20], [-1, 5, 9]]
(0) (1)

```
al.get (1) ⟶ [-1, 5, 9]
              0  1  2
```

	0	1	2
0	10	20	
1	-1	5	9

// access 9

```
al.get(1).get (2); ⟶ 9
[-1, 5, 9]
 0  1  2
```

(0, 1) → 20

(1, 2) → 9

// access 20 from al

```
al.get (0).get (1); ⟶ 20
[10, 20]
 0   1
```

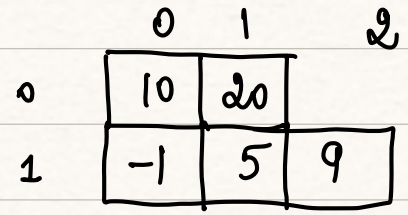
al.get(i).get(j)

arr[i][j]
row column

no. of rows → al.size(); ⟶ 2

no. of columns → al.get(0).size(); → 2

al.get(1).size(); → 3



change 9 to 19

al.get(1).set(2,19);

[-1, 5, 9]
0 1 2

al: [[10,20], [-1,5,19]]
0 1 2
0 1

how to remove 5

al.get(1).remove(1);
↳ index

al: [[10,20], [-1, 19]]
0 1

Ques. Row sum

Given a 2D ArrayList [having same no. of elements in every row], create an arrayList that stores the sum of each row

[[10, 20, 30] , [1, 2, 3]]

0 → [10, 20, 30]

1 → [1, 2, 3]

	0	1	2	
0	10	20	30	⇒ 60
1	1	2	3	⇒ 6

o/p → [60, 6]

mat[0][2]

mat.get(0).get(2)

Doubts

Matrix scalar product

$$A = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 2 & 5 \\ 4 & 7 & 8 \end{bmatrix} \qquad B = 2 \qquad \text{ans} - \begin{bmatrix} 2 & 4 & 6 \\ -2 & 4 & 10 \\ 8 & 14 & 16 \end{bmatrix}$$

create new 2D Array of same dimension

ans [i][j] = B x A[i][j];

{1, 2} , {-, -, -} , {-1-1-1-}

0 1 2

2, 3, 4

main ( ) {

A =

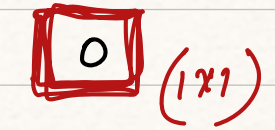
2	5	9	8
6	2	5	7
1	3	0	4

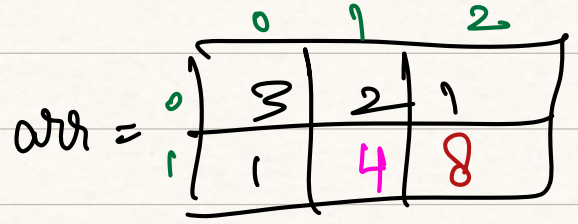
A = func (A); ⟶ A = [0] 1x1

S.O.P(A[0][0]); ⟶ 0

}

```
int[][] fun(int[][] mat)
    int[][] A = new int[1][1];
    return A;
}
```





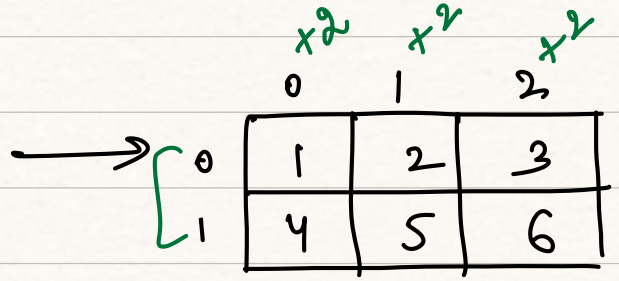
row = 1 col = 1
2

row = 1 ; row < 2 ; row ++

col = 1 ; col < 3 ; col ++

✓ if (odd) → +1

⇒ if (even) → x2



N x M

0th col sum

i j
0 , 0
1 , 0

0th row sum = 1+2+3 = 6

1st row sum = 4+5+6 = 15

0th col sum = 1+4 = 5

1st col sum = 2+5 = 7

2nd col sum = 2+6 = 9

ans = [6 , 15 , 5 , 7 , 9]
0 1 2 3 4

(N+M)

i j
(0,0) + (0,1) (0,2)

```
for (int i = 0, i < N ; i++ ) {
    int sum = 0;
    for ( int j = 0; j < M ; j++) {
        sum += arr[i][j];
    }
    ans[i] = sum;
}
```

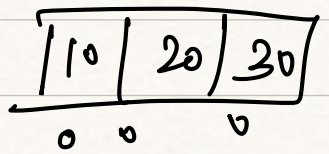
sum of ith row

0,0
0,1
0,2

```
for (int j = 0 ; j < M ; j++) {
    int sum = 0;
    for (int i = 0; i < N ; i++){
        sum += arr[i][j];
    }
    ans[N+j] = sum;
}
```

sum of jth col

arr = {10, 20, 30}



arr = new int [3];