2D ArrayList

ArrayList<Integer>

ArrayList<ArrayList<Integer>>

// Declaration

```
ArrayList<ArrayList<Integer>> mat;
```

// Initialisation

```
AL<AL<Integer>> mat = new AL<AL<Integer>>();
```

// Add operation mat :

```
AL<Integer> arr1 = new AL<Integer>();
arr1.add(1);
arr1.add(-3);
```

arr1: 1, -3

```
mat.add(arr1);
```

mat: 1, -3

```
AL<Integer> arr2 = new AL<Integer>();
```

```
arr2.add(0);
arr2.add(10);
```

arr2 : 0, 10

```
mat.add(arr2);
```

mat : 1, -3
0, 10

```
AL<Integer> arr3 = new AL<Integer>();
arr3.add(-4);
arr3.add(6);
arr3.add(5);
```

arr3 : -4, 6, 5

```
mat.add(arr3);
```

mat : 1, -3
0, 10
-4, 6, 5

Each arraylist within 2D arraylist can have different sizes

// Transpose

$$A = \begin{matrix} & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 3 & 2 \\ 2 & 6 & 0 \end{matrix}$$

$$A^T = \begin{matrix} & 0 & 1 & 2 \\ 0 & 1 & 3 & 6 \\ 1 & -1 & 2 & 0 \end{matrix}$$

- Rows become columns
- Cols become rows

$$A = \begin{array}{c} \\ 0 \\ 1 \end{array}\begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{bmatrix} 1 & 3 & 5 \\ 6 & -1 & 0 \end{bmatrix} \end{array} \qquad A^T = \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} 1 & 6 \\ 3 & -1 \\ 5 & 0 \end{bmatrix} \end{array}$$

$i \quad j \qquad\qquad i^T \quad j^T$

$A[0][0] = A^T[0][0]$

$A[0][1] = A^T[1][0]$

$A[0][2] = A^T[2][0]$

$A[1][0] = A^T[0][1]$

$A[1][1] = A^T[1][1]$

$A[1][2] = A^T[2][1]$

$i = j^T$

$j = i^T$

$A[i][j] = A^T[j][i]$

```
int[][] transpose (int[][] A){
    int n = A.length;
    int m = A[0].length;
    int[][] B = new int[m][n];
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            B[j][i] = A[i][j];
        }
    }
    return B;
```

```
}

ArrayList<ArrayList<Integer>> transpose(ArrayList<ArrayList<Integer>> A){

    ArrayList<ArrayList<Integer>> B = new ArrayList<ArrayList<Integer>>();
    int n = A.size();
    int m = A.get(0).size();
    for(int i=0; i<m; i++){
        ArrayList<Integer> arr = new ArrayList<Integer>();
        for(int j=0; j<n; j++){
            arr.add(A.get(j).get(i));
        }
        B.add(arr);
    }
    return B;
}
```

A:

1	2
3	4
5	6

B:

i	j	A[j][i]	arr	B
0	0	1	1	[ ]
0	1	3	1, 3	[ ]
0	2	5	1, 3, 5	[ ]
0	3	→ Breaks	—	[[1, 3, 5]]

1	0	2	2	"
1	1	4	2, 4	"
1	2	6	2, 4, 6	"
1	3	→ Breaks	—	[[1, 3, 5], [2, 4, 6]]

2 → Breaks

Identity

$a + 0 = 0 + a = a$

Additive identity

$a \times 1 = 1 \times a = a$

Multiplicative identity

$A * B = C$

$A * I = I * A = A$

Matrix multiplicative identity

$I_n \quad [n \geq 1]$

$$I_2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

I_1 [1]

$$I_4 \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

2D Arraylist Identity matrix - Size N

```
AL<AL<Integer>> identity(int N){

AL<AL<Integer>> I = new AL<AL<Integer>>();

for(int i=0; i<N; i++){
  AL<Integer> arr = new AL<Integer>();
  for(int j=0; j<N; j++){

    if(i==j){
      arr.add(1);
    }else{
      arr.add(0);
    }
  }
  I.add(arr);
}

return I;
}
```

Doubts

A: [1, 2, -1, 0, 5]
0 1 2 3 4

Y = 10
X = 2

A: [1, 2, 10, -1, 0, 5]
0 1 2 3 4 5

```
— fun( int[] A){

    n = A.length
    int[] ans = new int[n+1];

    for(int i=0; i<x; i++){
        ans[i] = A[i]
    }

    ans[x] = y;
    for( int i=x; i<n; i++){
        ans[i+1] = A[i];
    }
}
```