

$$\rightarrow \underset{\text{mat}}{A} \times \underset{\downarrow}{I} = \underline{A}$$

Identity matrix \rightarrow is a square matrix

$$N=3 \quad \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \quad \begin{matrix} 3 \times 3 \\ A \rightarrow \end{matrix} \quad \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

$$\underset{2 \times 3}{A} \times \underset{3 \times 3}{I} = \underline{A}_{2 \times 3}$$

$$\underset{4 \times 5}{A} \times \underset{5 \times 5}{I} = \underline{A}_{4 \times 5}$$

\rightarrow Given A return its identity matrix.

$$M = A[0].length; \quad N=3$$

int [][] id = new int [M][M]

for(int i=0; i<M; i++)
for(int j=0; j<M; j++)

if(i==j)
id[i][j]=1;

return id;

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

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

$$A \rightarrow \underline{3 \times 3}$$

\downarrow

$$A^T \rightarrow \underline{2 \times 2}$$

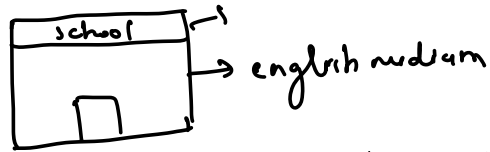
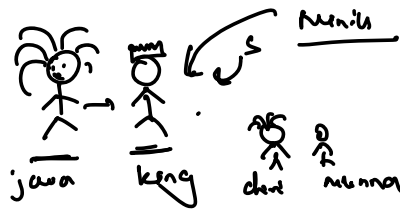
$$\underline{\text{ex}}: \begin{bmatrix} 2 & 3 & 1 \\ 4 & 5 & 6 \end{bmatrix}_{2 \times 3} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

\Downarrow

$$\begin{pmatrix} 2 \times 1 + 3 \times 0 + 1 \times 0 & 2 \times 0 + 3 \times 1 + 1 \times 0 & 2 \times 0 + 3 \times 0 + 1 \times 1 \\ 4 \times 1 + 5 \times 0 + 6 \times 0 & 4 \times 0 + 5 \times 1 + 6 \times 0 & 4 \times 0 + 5 \times 0 + 6 \times 1 \end{pmatrix}$$

$$\begin{bmatrix} 2 & 3 & 1 \\ 4 & 5 & 6 \end{bmatrix}$$

→ Give a matrix A and B check if B is identity matrix of A
 $A_{2 \times 3}$



```
int[] school = new int[10];
for(i=0 — 10) school[i]
arr
```

→ 3 new student

```
int[] school1 = new int[13];
for(i=0 — 10
  school1 ← school[i]
}
for(i=3)
  school1 ← 3 new stud.
```

again 5 students.

```
int[] school2 = new int[18];
for(i=0 — 13)
  school2 ← school[i]
}
for(i=13 — 18)
  school2 ← 5 new students.
```

10 Array Astra has fixed size (input) → dynamic

ArrayList → Dynamic size
 [10, 20] ∪ 30 [10, 20, 30] ∪ 40
 [10, 20, 30, 40]

Syntax

int[] arr = new int[];

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

↑
uppr case

↑
uppr case

↓
variable
↓
user defined

int[] arr = new int[10];

arr[0] = 1;

arr[1] = 2;

arr[2] = 20;

arr.length; 1 → 3

arr[3] = 30

0	1	2	3	4
1	2	20	30	4

20 → 21

arr[4] = 40

so p(arr[2]);

arr[2] = 21;

for(int i=0; i < arr.length; i++)

so p(arr[i]);

4

0	1	2	3	4
1	2	21	30	40

1) end

2) for (if index == 2 continue

ArrayList < Integer > al = new ArrayList();

al.add(1); → [1]

al.add(2); [1, 2]

al.add(20); [1, 2, 20]

0	1	2
1	2	20

al.size(); → 3

al.add(30);

0	1	2	3
1	2	20	30

al.add(40);

0	1	2	3	4
1	2	20	30	40

al.get(2); 11 → 20 21

Redefined function

0	1	2	3	4
1	2	21	30	40

al.set(2, 21);

index ← value.

so p(al); [1, 2, 21, 30, 40]

so p(al);

al.remove(2);

index

0	1	2	3
1	2	40	30

19th Oct 10:00 am
↓
21st Oct 8:00 pm

90 mins — TRANSITION TEST → Mandatory

pass → Ds. algot. P.s
↓
4 →
7

Rest of the information
will be shared
via Email

Break → 10:35 → 10:45
10:45

Quiz

1)

list.size();

2)

ArrayList<Integer> list = new ArrayList<>();

list.add(10);

list.add(20);

list.add(30);

System.out.println(list.get(1));

0 1 2
10 20 30
↓

20

2)

```
ArrayList<Integer> list = new ArrayList<>();
list.add(10);
list.add(20);
list.add(30);
list.add(16);
list.remove(2);

sop(list);
```

0	1	2	3
10	20	30	16

0	1	2
10	20	16

[10, 20, 16]

2D ArrayList

```
int[][] arr = new int[5][5];
```

2D-array → arrays inside array
↳ array of arrays.

2D-ArrayList → ArrayList of ArrayLists
↳ Collection ArrayList

```
ArrayList<Integer> al = new ArrayList<>();
al.add(20);
```

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

~~mat.add(20);~~ // error

```
mat.add(al);
mat.add(al);
```

```
ArrayList<Integer> list = new ArrayList<>();
list.add(20);
list.add(30);
```

[20, 30]

```
mat.add(list);
```

[[20, 30]]

```

ArrayList<Integer> list2 = new ArrayList<>();
list2.add(1);
list2.add(3);
list2.add(5);

```

[1, 3, 5]

```

mat.add(list2);

```

[[20, 30],
[1, 3, 5]]

outer arraylist

[[20, 30], [1, 3, 5]]

inner arraylists.

```

ArrayList<Integer> list3 = new ArrayList<>();

```

```
list3.add(40);
```

```
list3.add(50);
```

```
list3.add(60);
```

```
list3.add(70);
```

```
list3.add(80);
```

[40, 50, 60, 70, 80]

```
mat.add(list3);
```

[[20, 30],
[1, 3, 5],
[40, 50, 60, 70, 80]]

0 1 2 3 4
0 [20, 30]
1 [1, 3, 5]
2 [40, 50, 60, 70, 80]

0 1 2 3 4

[0] 10-3

```
mat.get(2).get(4);
```

80

`rows = mat.size() / 3;`
`mat.get(0).size(); // 2`
`mat.get(1).size() // 3`

`[`
`[20, 30]`
`[1, 3, 5]`
`[40, 50, 60, 70, 80]`
`]`

`[`
`[20 30]`
`[1 3 5]`
`[40 50 60 70 80]`
`]`
 ↓
 2D arraylist ✓
 X → Matrix.

X THE - END X

Doesn't

`prum(> 1`

`int 0(0) = 1`

`A = func(A);`
`solve(A[0][0]);`

`[`
`2 5 9 8`
`1 3 0 4`
`6 2 5 7`
`]`

`[`
`0`
`]`

`solve(int[][] mat)`

`int[][] A = new int[1][1];`

`return A;`
`mat = new int[3][3];`
`return`

1) A ↪

2) eA, oA; eindex=0, oddindex=0

2) for(int i = 0; i < n; i++)

if(A[i] % 2 == 0) {
eA[eindex] = A[i];
eindex++;

else {

oA[oindex] = A[i];

oindex++;

