→ 10th SSC results came :

↳ 6 exams :

```
int e1, e2, e3, e4, e5, e6;

Scanner sc = new Scanner(System.in)
e1 = sc.nextInt();
e2 = sc.nextInt();
e3 = sc.nextInt();
e4 = sc.nextInt();
e5 = sc.nextInt();
e6 = sc.nextInt();
```

Total Sum: $e_1 + e_2 + e_3 + e_4 + e_5 + e_6$

Avg : $\frac{e_1 + e_2 + e_3 + e_4 + e_5 + e_6}{6}$

→ B-tech / Degree :

↳ 50 exams

```
int e1 e2 e3 e4 .. e50;

e1 = sc.nextInt();
e2 = sc.nextInt();
e3 = sc.nextInt();

e50 = sc.nextInt();
```

Total sum = $e_1 + e_2 + e_3 + \dots e_{50}$

→ The above is not efficient

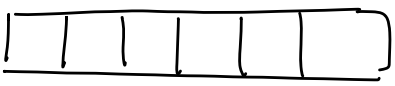
// Arrays: → { Same type of data }

int x; → int variable declaration → { 1 value }

// way-1

```
int ar[];   // declaring
ar = new int[6]; // initializing space
```

↳ 6 integers

ar → 

// way-2 → { Declaring & Initializing space }

```
{ int ar[] = new int[10]; }
```

→ { // generaly prefer }

↳ 10 integers

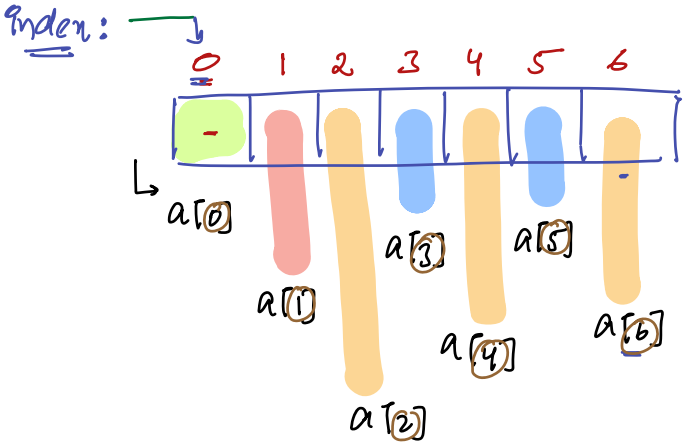
// way-3 → { Declaring & Initia space }

```
int[] ar = new int[100];
```

↳ 100 integers

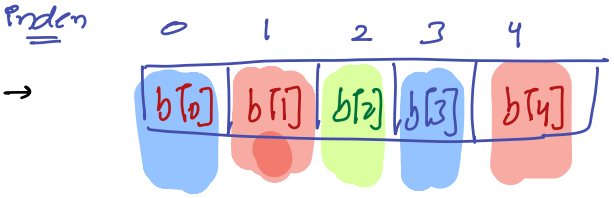
// int ar[10] ✗ → { won't work in java }

```
// int a[] = new int[7];
```




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

→




```
int c[] = new int[100]
```

→ 1st Block → c[0]

↓

100th Block → c[99]

```
int d[] = new int[N]
```

→ 1st Block → d[0]

↓

Nth Block → d[N-1]

↳ Index: [0, N-1]

Note: Indexing start at 0 in arrays