Data

raw, not useful
random

Information

organised data, useful

Ques. marks of 5 students find avg.

```
int s1 = scn.nextInt();
int s2 = scn.nextInt();
int s3 = scn.nextInt();
int s4 = scn.nextInt();
int s5 = scn.nextInt();

int sum = s1 + s2 + s3 + s4 + s5;

double avg = sum / 5.0;
```

60 students

S1
S2
S3
S4
⋮
S60

sum = S1 + S2 + - - - - - + S60

avg = $\frac{sum}{60.0}$;

Arrays → Data structure

collection of same type of data

```
int n = 10;
```

→ 1 integer space

10
n

int

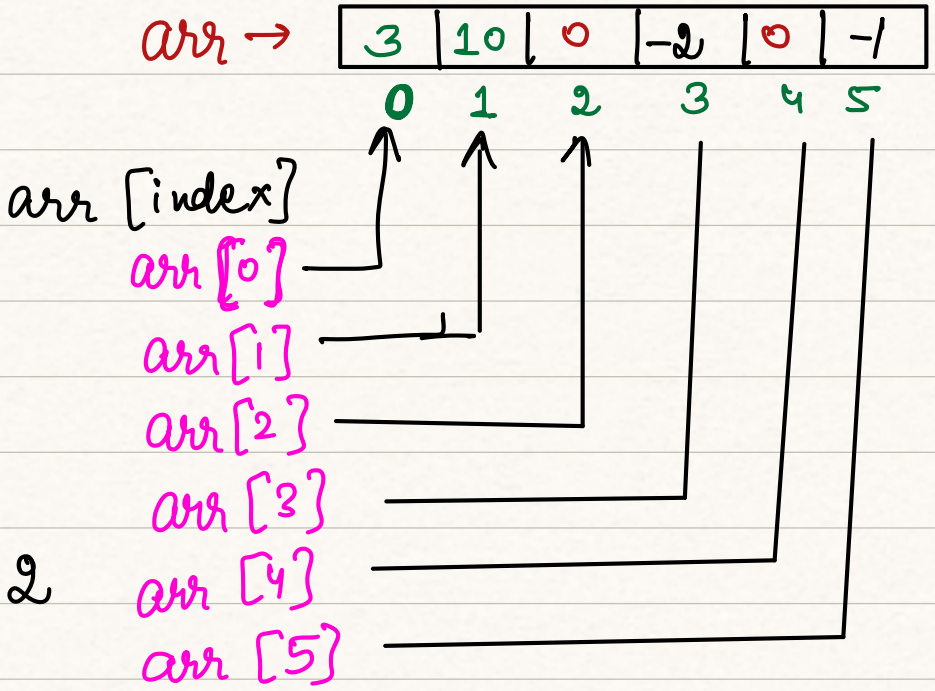
```
int [] arr = new int [5];
```

- int → data type
- arr → name
- 5 → size → 5 integer spaces

Ques. create an array that can store 6 integer values.

```
int [] arr = new int [6];
```

```
int arr[] = new int[6];
S.O.P(arr[3]); // 0
```



indexes

```
arr[0] = 3;
arr[1] = 10;
arr[3] = -2;

S.O.P(arr[3]); // -2

arr[6] = 15; // Error
```

↳ Array index out of bounds

```
arr[5] = -1;

SOP(arr[-1]); // Error
```

index ⟶ 0 ⟶ n-1

S.O.P (arr[n]); //Error

size = n

Ques. create an array of 5 size and then take input from user of 5 integer values store them in array and print them.

I/P → 10 20 5 9 7

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

arr [0] = scn.nextInt();
arr [1] = scn.nextInt();
arr [2] = scn.nextInt();
arr [3] = scn.nextInt();
arr [4] = scn.nextInt();
```

```
int index = 0;
arr[index] = scn.nextInt();
index++;
arr[index] = scn.nextInt();
index++;
arr[index] = scn.nextInt();
index++;
arr[index] = scn.nextInt();
index++;
arr[index] = scn.nextInt();
index++;
```

// arr[0] → 1

// arr[1] → 2

// arr[2] → 3

// arr[3] → 4

// arr[4]

i <= 4

i < 5

```
for (int i = 0; i < 5; i++) {
    arr[i] = scn.nextInt();
}
```

I/P → 10 20 5 9 7

10	20	5	9	7
0	1	2	3	4

i	i < 5	arr[i]	i++
0	true	arr[0]	1
1	true	arr[1]	2
2	true	arr[2]	3
3	true	arr[3]	4
4	true	arr[4]	5
5	false →		Break

```
for (int i=0 ; i < 5; i++) {
    S.O.P (arr[i]);
}
```

Highest marks scored [maximum element of an array]

```
int[] marks = {1, 15, 21, 100, 17, 05}
               0   1   2   3   4   5

int n = marks.length; // n = 6        ans = 100

int max = 0;

for(int i=0; i<n; i++){
    if (marks[i] > max){
        max = marks[i];
    }
}

S.O.P (max);
```

marks = {1, 15, 21, 100, 17, 85}
0 1 2 3 4 5

max : 0

i	i < 6	marks [i] > max	max	i++
0	true	1 > 0 (true)	1	1
1	true	15 > 1 (true)	15	2
2	true	21 > 15 (true)	21	3
3	true	100 > 21 (true)	100	4
4	true	17 > 100 (false)	100	5
5	true	85 > 100 (false)	100	6
6	false	→	break	

max = 100

Quiz 1. array containing 5 int values

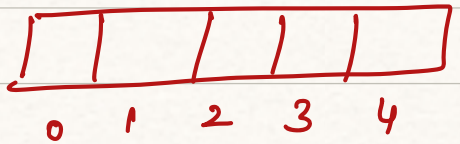
```
int [] arr = new int [4];   x
```

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

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

Quiz 2.

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

int sum = 0;

for (int i = 0; i < 6; i++) {
    sum += arr[i];
}

S.O.P (sum);
```



i < 6

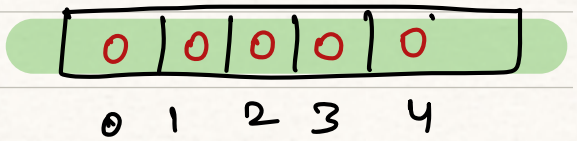
i = 0 1 2 3 4 5

arr [5] // Error

Array Index Out of Bounds

Quiz 3.

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

int sum = 0;

for (int i = 0 ; i < 5 ; i ++) {
    sum += arr[i] ;
}

S.O.P (sum);
```



0 + 0 + 0 + 0 + 0

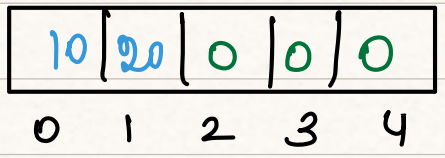
→ // 0

Quiz 4.

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

    arr [0] = 10;
    arr [1] = 20;

    int sum = 0;

for (int i = 0 ; i < 5 ; i ++) {
        sum += arr[i] ;
}

S.O.P (sum);
```

10	20	0	0	0
0	1	2	3	4

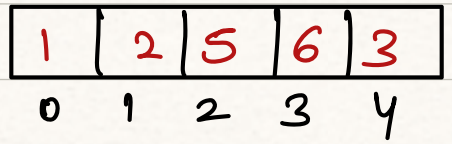
$10 + 20 + 0 + 0 + 0$



$\longrightarrow$ // 30

Quiz 5.

```
int [] arr = {1, 2, 5, 6, 3};

int sum = 0;

for (int i = 0; i < 5; i++) {
    sum += arr[i];
}

S.O.P (sum);
```



// 1+2+5+6+3

→ // 17

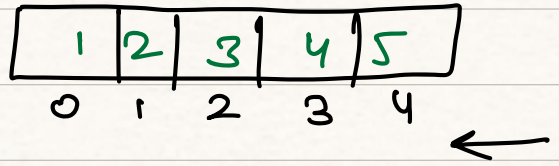
Quiz. what is the last index of an array of size N

N-1

Print in reverse

int[] arr = { 1, 2, 3, 4, 5 }

O/p → 5 4 3 2 1




```
int n = 5;
for ( int i = n-1; i >=0 ; i-- ){
        S.O.P (arr [i]);
}
```

i ≥ 0

i = ~~4~~ ~~3~~ ~~2~~ ~~1~~ ~~0~~ -1 ✗

O/p → 5 4 3 2 1