

ARRAYS

Non Primitive datatypes



build using primitive datatypes

or non primitive

① ARRAYS

```
int m1 = 30;  
int m2 = 40;  
int m3 = 50;
```

30	40	35	50	60
0	1	2	3	4

int marks []; // Declaration int

reference variable marks = new int [5]; // Initialisation

⇒ marks[0] = 30; 1st
⇒ marks[1] = 40; 2nd
⇒ marks[2] = 35; 3rd
marks[3] = 50; 4th
marks[4] = 60; 5th

Size of array // length
→ marks of students

Print marks

```
for(int i=0; i < marks.length; i++) {  
    sys (marks[i]);  
}
```

MEMORY MANAGEMENT IN ARRAY

```
int marks[];
```

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

allocates the
memory
in heap

```
int a = 5;
```

$$\text{marks}[2] = 20$$

```
int arr[]; // int [] arr;
```

Boolean array

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

Memory
RAM

a	5
manab	4k

Task

$$4k+4 \quad 4k+12$$

0	0	20	0	0
---	---	----	---	---

$$\begin{array}{r} 4k \\ 4k+8 \\ \quad +16 \\ \hline 4k \end{array}$$

20 | 30 | 40 | 50

→ `int arr[] = { 20, 30, 40, 50 };`

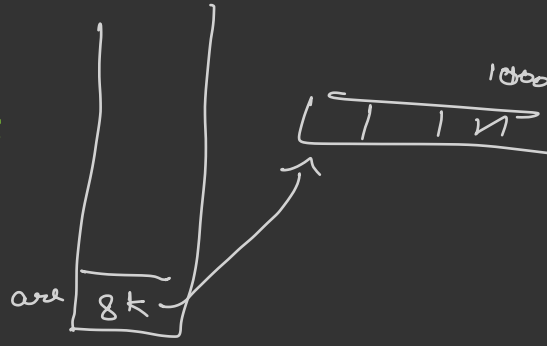
`int arr[] = new int [4];`
`arr[0] = 20;`
`arr[1] = 30`
`arr[2] = 40`
`arr[3] = 50`

expanded

`char arr[] = new char [2];`

`int arr = new int [1000];`

Same time { `arr[500] = 20;`
`arr[0] = 10;`



$8K + 4 \times 500$
 $8K + 2K$
 $= 10K$

10000

```
int a[] = new int [3];
```

```
a[0] = 0;
```

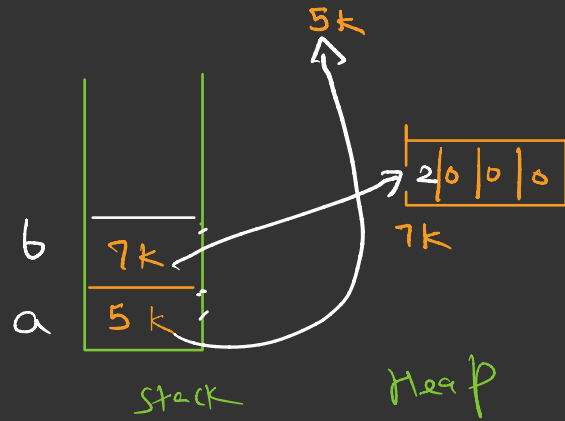
```
a[1] = 1;
```

```
a[2] = 2;
```

```
int b[] = a;
```

```
b[1] = 3; ←
```

0	1	2
0	3	2



```
Print a 0 3 2
```

```
Print b 0 3 2
```

```
→ b = new int [4];  
→ b[0] = 2;
```
