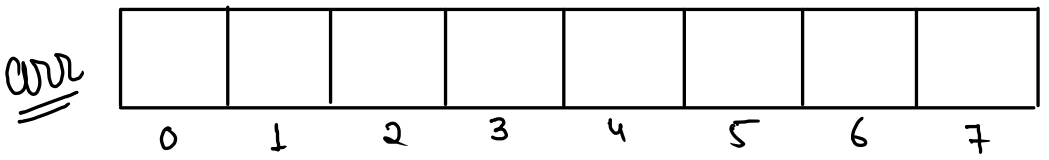
⇒ 1D array (collection of similar type of data type)

- String
- int
- boolean
- float
- char
- double



size = 8

indexing

size = arr.length

last index = size - 1

Keywords

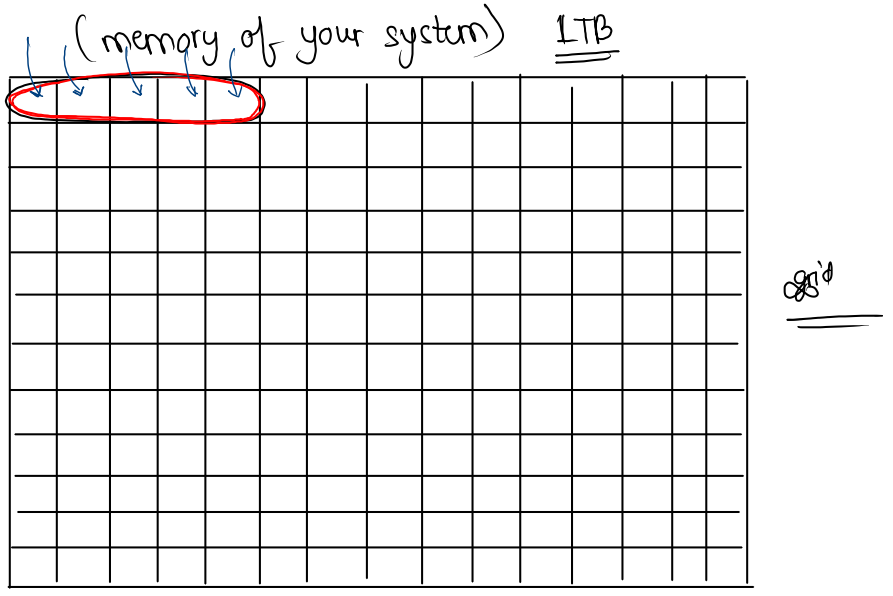
↳ declare 1D array

number (int)

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

data_type [] arr_name = new data_type[size];

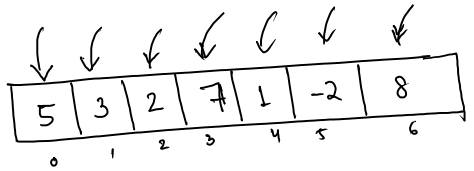
Note:- array will do continuous memory allocation



↳ max of array

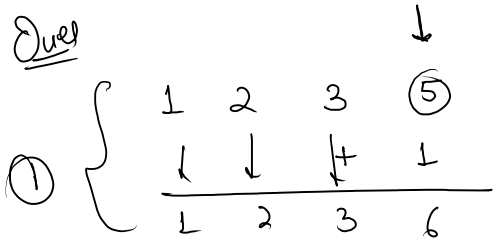
```java
// main logic
public static int maximumValue(int[] arr, int n) {
    int val = Integer.MIN_VALUE;
    for (int i = 0; i < n; i++) {
        if ( arr[i] > val ) {
            val = arr[i];
        }
    }
    return val;
}
```

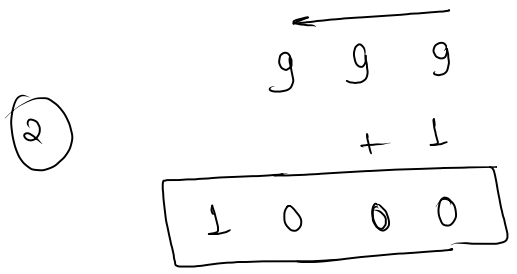
satisfied → update



val = ~~-∞~~ ~~5~~ ~~7~~ 8 ←

Ques





1 1
1-8

1000

steps
1) handle 1-8 by +1
2) 9 → keep placing 0 until 9 is there.
and use carry.

ex = 29299

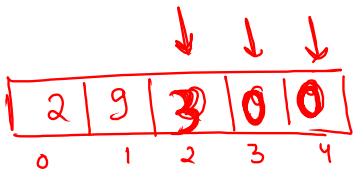
29300

```java
public static int[] plusOne(int[] arr, int n) {
    for (int i = n - 1; i >= 0; i--) {
        if ( arr[i] < 9 ) {
            arr[i] = arr[i] + 1;
            return arr;
        }
        arr[i] = 0;
    }
    int[] ans = new int[n + 1];
    ans[0] = 1;
    return ans;
}
```

2<9

29200
29201

arr =



(29299)

ans 2 9 3 0 0

```java
public static int secondLargest(int[] arr, int n) {
    Arrays.sort(arr);
    for (int i = n - 2; i >= 0; i--) {
        if ( arr[i] != arr[i + 1] ) {
            return arr[i];
        }
    }
    return Integer.MIN_VALUE;
}
```

↳ what if we cannot sort it and we can traverse only once in array } follow up