Memset ( ) function :-

⇒ used to fill a block of memory with a particular value.

(0 / -1)

Syntax : void * memset (void * ptr , int value , size_t num)

↳ ~~sets~~ [REFER TO ~~GI~~ DSA notes]

WEEK 3
LECTURE 2

VECTORS

VECTOR
- → Data structure
- → Same as array , but dynamic
  - ↳ no fixed size.
- → default size = 0
- → if it gets full , then new items are inserted and the size gets doubled.
- → pass by value in functions.

Concern : the concept of doubling the size of vector can lead to memory wastage.

Initialisation :

```
vector <int> arr{10,20,30};   →   [10|20|30]
vector <int> arr(5);   →   [0|0|0|0|0]
vector <int> arr (5, -2);   →   [-2|-2|-2|-2|-2]
                  ↙    ↘
               size    value

int n ;  cin>>n; → let n=5

vector <int> arr(n);   →   [0|0|0|0|0]
vector <int> arr (n,10);   →   [10|10|10|10|10]
```

Insertion :

arr.push_back(5);

Deletion:

vector <int> arr ;

$$\begin{tikzcd} \ \arrow[r] \arrow[d, dash] & \text{arr.size()} \rightarrow 0 \\ \ \arrow[r] & \text{arr.capacity()} \rightarrow 0 \end{tikzcd}$$

Remove :

arr.pop_back (5);

Size : arr.size (); → no. of elements it stores

Empty or not : arr.empty(); → true, if empty.

Capacity : arr.capacity(); ⟶ * by 2, if array gets fully filled and a new element is then inserted.

⟶ no of elements it can store.

⟶ In initialisation, capacity = size in all methods of initialisation.

Q: Find the unique element in array. Every element occurs twice except one element.

i/p : { 1, 2, 4, 2, 1, 4, $\boxed{3}$ }
↑occuring only once

o/p : 3

[using XOR operator]

XOR → cancels out same element.

0^ ans = ans
↪ 0^1 = 1
↪ 0^0 = 0

Tip : for output of array:

```
for (auto value : arr) {
        cout << value << "__";
}
```