

8d - Arrays

Thursday, 13 August 2026 04:00

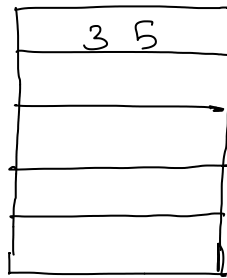
Data structure which is a collection of data items of same data type.

```
int a[10];
```

Array is continuous.

eg:

```
int number[5];  
number[0] = 35
```



number[0]
" [1]

" [5]

Array Initialise:

① `int myArr[5] = { 1, 1, 1, 1, 1 };`

All initialised as 1

② `int myArr[5] = { 1, 2 };`

[0] = 1, [1] = 2, rest are 0

③ `int myArr[5] = { 0 };`

All element 0

④ `static int myArr[10];`

All element is 0 since it's a global

array

⑤ `int myArr[10];`

All values are garbage.

Character Arrays

If something isn't initialised, we get `\0`
i.e. null character

```
1. char t[5] = {'a', 'b', 'c', 'd'}; // initialize to 'a','b','c','d','\0'
2. char t[4] = {'a', 'b', 'c', 'd'}; // initialize to 'a','b','c','d'
3. char s[5] = "abcd"; // initialize to 'a','b','c','d','\0' (short form of 1st)
4. char s[4] = "abcd"; // initialize to 'a','b','c','d' (short form of 2nd)
```

`char t[7] = "Godless"`
"

`char t[7] = { 'G', 'O', 'D', 'L', 'E', 'S',
 's', 's' }`

Null `char s[3] = { 'a', 'b' }`

$\therefore s[2] = \text{\0}.$

No dimension specified - NULL Character

Dimension taken from initialization.

```
int myArray[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

```
char s[] = "abcd"; // equivalent to char s[5] = "abcd";
```

→ Compiler puts a NULL at the end for char arrays if size not specified

```
char c[] = {'a', 'b', 'c', 'd'}
```

→ Initialisation like this does not add NULL.