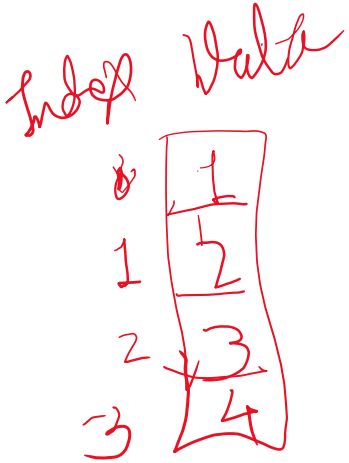
# Let’s solve it

# Initialization of arrays

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




```
int arr[5] = {1};
```


remaining are initialized to zeroes.

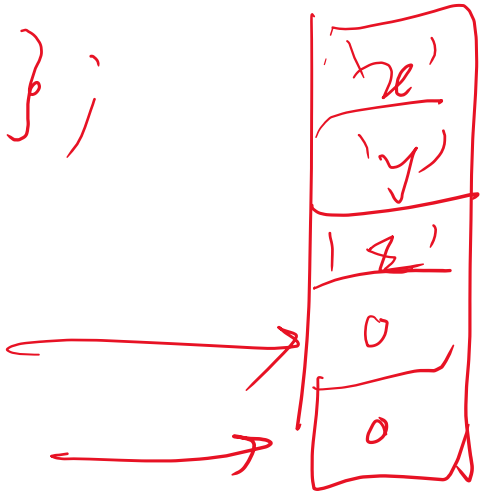
# character arrays

```
char arr[5] = {'a', 'c', 'f', '0', 'x'};
char arr[] = {'a', 'b', 'c'};
```

Each element of character array is going to be a character byte

```
char arr[5] = {'x', 'y', 'z'};
```

character 0 is having a name NULL



# Ascii table

storage symbol

NULL

0
1
2
3

| | |
|---|---|
| 65 | 'A' |
| ⋮ | |
| 90 | 'Z' |

| | |
|---|---|
| 97 | 'a' |
| ⋮ | |
| 122 | 'z' |

| | |
|---|---|
| 48 | 0 |
| 49 | 1 |
| ⋮ | ⋮ |
| 57 | 9 |

'0' 48

'\0' 0

NULL

#define NULL 0

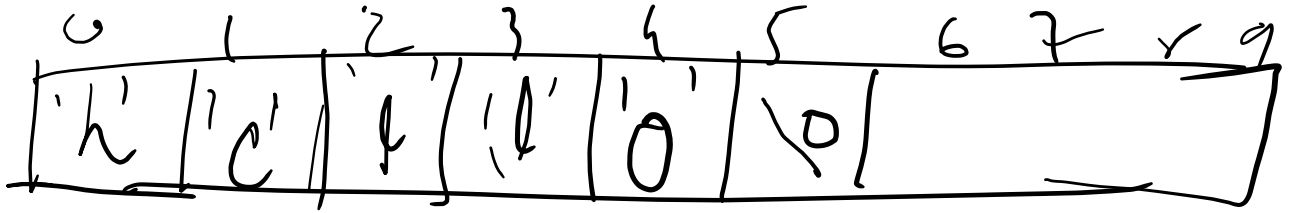
# Strings in C

constant/literal
"hello"

String as a variable
set of symbols followed by special NULL

string is stored as array of character arrays.
But character array not having NULL to indicate end is NOT a string.

```
char arr[10] = "hello";
```

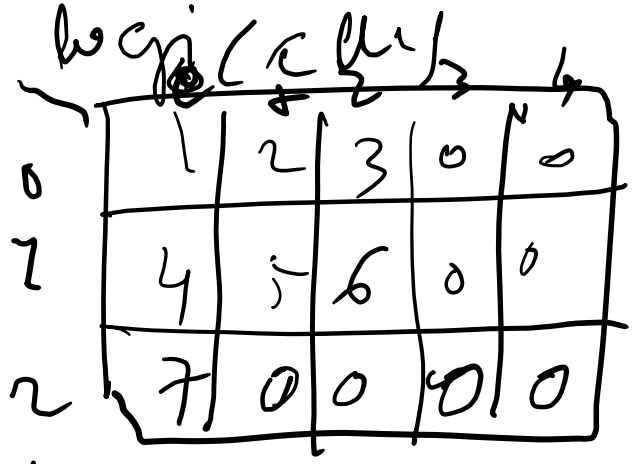



```
char arr[10] = {'h', 'e', 'l', 'l', 'o', '\0'};

printf("%s", arr);

puts(arr);
```

Two dimensional arrays

```
int arr2d[3][5];
```

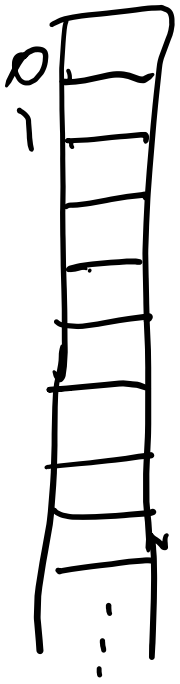
rows  cols



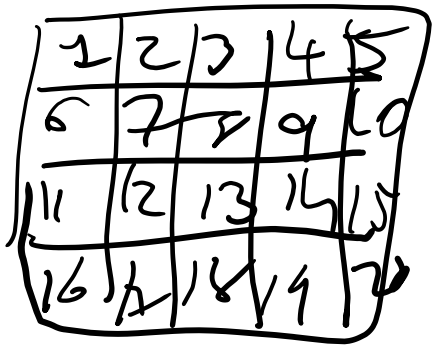

Initialization

```
int arr2d[3][5] =
    { {1,2,3},
      {4,5,6},
      {7} };
```

physically



Ex: Create below




```
int arr[4][5];
int count=1;
for(r=0; r<4; r++)
{
    for(c=0; c<5; c++)
    {
        arr[r][c] = count++;
    }
}
```

} Runtime Initialization (assignment)

# Addition of two matrices 3×3

```
int a[3][3], b[3][3], result[3][3];
```

Read a

```
for(i=0; i<3; i++)
{
  for(j=0; j<3; j++)
  {
    scanf("%d", &a[i][j]);
  }
}
```

Read b

```
&b[i][j]
```

add

```
result[i][j] = a[i][j] + b[i][j]
```