C-12 ⇒ Strings in C

Part - 1

(Introduction to strings (Initialize string at Compile time)

String:

* string is simply array of characters. or one dimensional character array.
* Datatype is "char".

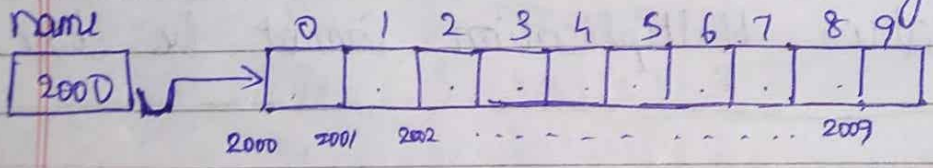
array ⇒ datatype arrayname[size]

string ⇒ char stringname[size]

* string is a character array which ends with null character.

Eg:- char name[10];

10 bytes memory
char size ⇒ 1 byte
∴ 1×10 ⇒ 10 bytes.



* We can store here only 9 characters with null character.

* Format specifier for string is %s

Strings :- → array of characters.

* Whatever we put inside double quotes is considered as string. (" ")

* Whatever we put inside single quotes is considered as character. (' ')

Eg: "Jenny lectures" "12345" "$abc"

'$' '1' 'a' 'A' "12"

* But how values of char will be stored in memory. ASCII values will be stored for equivalent character values.

Eg: char a = '1';
printf ("%c", a); → 1
printf ("%d", a); → 49.

For '1' Equivalent ASCII value is 49.

Eg: char a = 'A';
printf ("%c", a); → A
printf ("%d", a); → 65

For 'A' Equivalent ASCII value is 65.

Likewise we have 256 characters; so we have 256 possibles and in memory takes 1 byte = 8 bits to store character.

$2^8 = 256$, (0-255)

Eg:- Char C = 265;

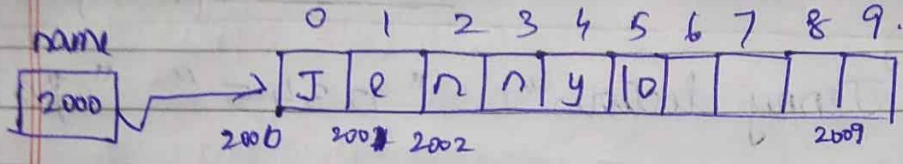
printf ("%d", c); → 9

printf ("%c", c); → -

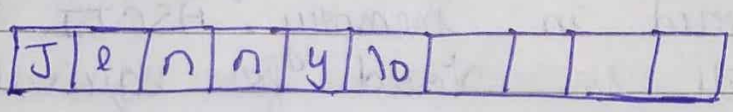
265
256
9

* String is put in double quotes (i.e) " "
String literal.

correct Eg:- Char name[10] = "Jenny";

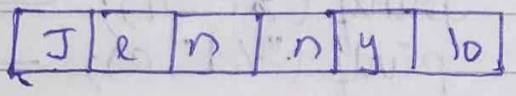


correct Eg:- Char name[10] = {'J', 'e', 'n', 'n', 'y', '\0'};



correct Eg:- Char name[] = {'J', 'e', 'n', 'n', 'y', '\0'};

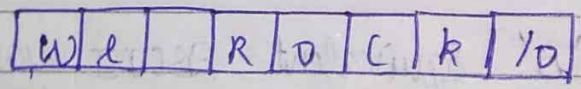
no size



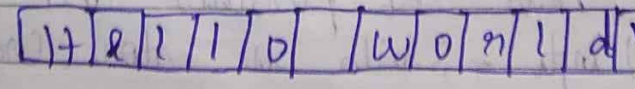
6 bytes allocate

In this case compiler will detect the size and here no wastage of size.

correct Eg:- Char name[] = {'W', 'e', ' ', 'R', 'O', 'C', 'K', '\0'};



correct Eg:- Char name[] = {"Hello World"};



Eg: char name[5] = {'w', 'e', ' ', 'R', 'O', 'C', 'K'};

Size 5

✗ wrong

Eg: Char name[10];
name[10] = "Jenny";

We cannot separate declaration and initialization.

✗ wrong

Eg: char name[10] = "Jenny";
Char s1[10];
s1 = name; ✗
s1 is array name; so we cannot copy array name into variable name.

✗ wrong

Eg: char a[] = "Hello world";
int a = 10;
array name ≠ variable name.

* Format specifier for string
%.s → string format specifier.