2.3 Strings.

* String is a sequence of characters terminated by a "null character '\0'".
* Strings in C are stored as an array of characters.

Syntax:- `char string-name[size];`

- `char` → data type
- `string-name` → name of string variable
- `[size]` → size of array.

* NOTE:- '\0' => used to indicate the termination of a string (that differs strings from normal character arrays)

Ways to initialize a String in C:-

① Assigning a string literal without size:-

```
char str[] = "C Programming";
// sizeof(str) => 13 (including '\0')
```

② Assigning a string literal with a predefined size:-

```
char str[50] = "C Programming";
```

=> Remember that we should always account for one extra space for '\0'.

```
// sizeof(str) => 50
```

③ Assigning character by character with size:-

```
char str[13]={'c','p','r','o','g','r','a','m','m','i','n','g','\0'};
```

Note:- '\0' should be added at the end.

④ Assigning character by character without size:-

```
char str[]={'c','p','r','o','g','r','a','m','m','i','n','g','\0'};
```

//sizeof(str) ⇒ 13.

Note: Size of the string is determined by the compiler automatically.

⇒ When a sequence of characters enclosed in the double quotation is encountered by the compiler, '\0' is appended at the end of the string by default.

Example:-

```
#include<stdio.h>
#include<string.h>
void main()
{
  char c1[]="Cprogramming";
  char c2[50]="Cprogramming";
  char c3[13]={'c','p','r','o','g','r','a','m','m','i','n','g','\0'};
```

```
char c4[] = {'c', 'p', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g', '\0'};
printf(" Size (c1) = %d \n Size (c2) = %d \n Size (c3) = %d \n
        Size (c4) = %d \n", sizeof(c1), sizeof(c2), sizeof(c3), sizeof(c4));
}
```

(2)

Output :-

Size (c1) = 13

Size (c2) = 50

Size (c3) = 13

Size (c4) = 13

Reading string input from user :-

① Using "%s" :-

```
#include<stdio.h>
void main()
{
  char s[50];
  printf("String = ");
  scanf("%s", s);
  printf("%s", s);
}
```

Output :-

① String = Hello
Hello

② String = Dennis Ritchie
Dennis

↓

content after whitespace is ignored.

Methods to avoid ignorance of string after whitespace:-

(1) Using gets():

Use gets() to read characters from the standard input (stdin) and store them as a C string until a newline character is reached.

```
#include<stdio.h>
void main()
{
  char name[30];
  printf("Name= ");
  gets(name);
  printf("Name entered = ");
  puts(name);
}
```

Output :-

```
Name = Dennis Ritchie
Name entered = Dennis
               Ritchie
```

(2) Using scanset:

scanset "%.[^\n]s" can be used to read string and store them in an array until newline (\n) is encountered.

```
#include<stdio.h>
void main() {
  char str[20];
  printf("Enter value= ");
  scanf("%.[^\n]s", str);
  printf("O/p: %.s", str);
}
```

Output:-

```
Enter value = Hi Hello
O/p: Hi Hello
```

Why no '&' before string-name in scanf?

```
char str[20];
scanf("%s", str);
```

* Ampersand (&) symbol must be used to provide the address of the variable to scanf() inorder to store the value read in memory.
* As "str[]" is a character array, using "str" will give the base address of the string.
* Hence using "str" implies that we are already providing the base address of the string to scanf.

Wrong initialization of string:-

```
Char s[10];
s = "HelloWorld";
```