The most basic definition of a string is that it is an array of characters. This definition is also true for see but there is a slight difference between a string and a character array, i.e., a string ends with a special character called the null character (\0). This character indicates the termination of a string. A string is declared in a very similar manner to a one-dimensional array,

```
char str_name(size);
```

It has three main parts,

a. `char`: as string is an array of characters it's data type has to be char.
b. `str_name`: It must be a valid name and should follow all the principles of declaring a variable name.
c. `size`: It is a constant value and defines the maximum capacity of a string.

Initialization of a string is similar to that of any other data type like int and float,

```
Char str (50);
printf("\nEnter the string: ");
scanf("%s",str);
```

NOTE:

The initialization will vary if we are trying to input a sentence. The problem is that scanf will stop reading whenever a white space or a newline or end of line (EOF) is reached. To solve this problem we have gets which does not stop reading when a white space is encountered. Hence whenever we want to input a sentence we will use gets.

Although string is an array we do not need to use any for loop for declaring it as scanf has an access specifier for strings. The only difference between the initialization of int and float and that of string is that we do not have to use '&' for strings. The reason is that, unlike other data types, when we use str it is as a reference to the memory location which was the use of '&'.

In C we have some predefined functions related to strings. To use them we have to include the header file string.h in the beginning like this,

```
include<string.h>
```

The following is a table of these functions:

Function	**What It Does**
strcmp(char str1, char str2)	Compares str1 and str2 in a case-sensitive way. If the strings match, the function returns 0.
strncmp(char str1, char str2,n)	Compares the first n characters of str1 and str2, returning 0 if the given number of characters match.
strcasecmp(char str1, char str2)	It is similar to strcmp the only difference is that it ignores the case.
strncasecmp(char str1, char str2,n)	It is similar to strncmp the only difference is that it ignores the case.
strcat(char str1, char str2)	Appends one string to another, creating a single string out of two and it is stored in str1. For example, strcat("for", "example") results in forexample.
strncat(char str1, char str2,n)	Appends n number of characters from str2 to the end of str1. For example, strcat("for", "example",3) results in forexa.
strchr(char str1, char c)	Searches for c within str1. The function returns c's position from the start of str1 as a pointer.
strrchr(char str1, char str2)	It is similar to strchr but it returns the character's position from the end of the string as a pointer.
strstr(char str1, char str2)	Searches for str2 inside str1. The function returns a pointer to str2's location if it is found.

strnstr(char str1, char str2,n)	Searches for str2 within the first n characters of the str1. The function returns a pointer to str2's location if it is found.
strcpy(char str1, char str2)	Copies str2 to str1.
strncpy(char str1, char str2)	Copies n number of characters from str2 to str1.
strrev(char str1)	Reverses str1 and stores in str1 itself.
strlen(char str1)	Returns the length of str1, not counting the NULL character at the end of the string.

PROGRAMS

Q. Write a program to take a word as an input from the user and check if it is palindrome or not.

https://youtu.be/Bd45YMcChO4

Q. Write a program to find the first occurrence of a word in a sentence where both the word and the sentence is entered by the user.

https://youtu.be/uW0rj9wGqM4

QUESTIONS ON STRINGS (PLS DO SOLVE THEM): - https://youtu.be/5TDOzO7H5DI