# C_68 ⇒ Program to Compare 2 strings

* Predefined function to compare the two strings is strcmp().

* It will compare the two strings character by character until the null character.

S1 ⇒ C O M P U T E R
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
S2 ⇒ C O M P U T E R
⇒ strings are equal

S1 ⇒ C P M P U T e R
↓ ↓ ↓ ↓ ↓ ↓ ↕ ↓
S2 ⇒ C P M P U T E R
⇒ strings are not equal

* Meaning of strcmp() is defined in string.h header file, because compiler has to understand by reading this function.

* strcmp() should returns strings are equal or not.

```
int strcmp(const char *str1, const char *str2)
{
    0 → when both strings are same
    +ve →
    -ve →  [ASCII value of character comparison returns in -ve or +ve value.]
}
```

a=97 b=98 c=99
A=65 B=66 C=67

Eg:

Str1 computer lo
Str2 Computer lo

int strcmp(const char *str1, const char *str2)
{
0 ⇒ same

+ve ⇒ when the first character that does not match is greater in ASCII value in string 1.

-ve ⇒ when the first character that does not match is lesser in ASCII value in string 1

}

* Constant is only constant to string 1 and string 2; but not constant to pointer.

*Str1 → * is pointer to string 1

Example program (1) (using strcmp() function)

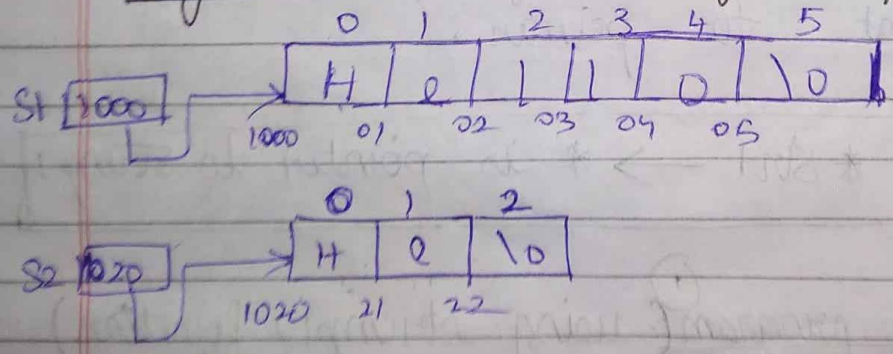
* To compare two strings.

```
#include <string.h>
void main()
{
  int value;
  char s1[] = "Hello";
  char s2[] = "welcome";

  value = strcmp(s1, s2);
  if (value == 0)
    printf ("strings are same");
  else
    printf ("strings are not same");
```

```
printf ("%d", value);
```
→ to see what value it returns

```
// if (strcmp(s1,s2) == 0)
```

Program ② (without using strcmp() function)



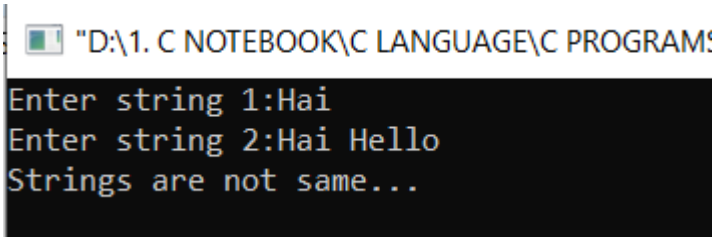

* If any one of the string has null character; then the comparison ends and prints result.

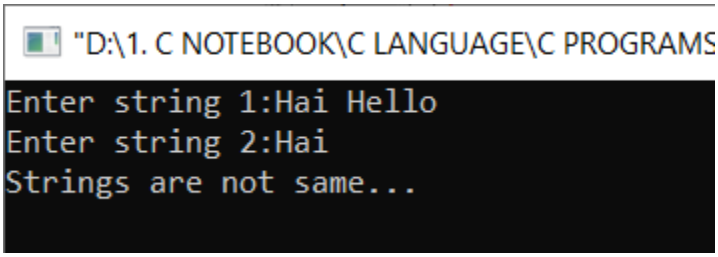
```
int i, flag=0;
Char S1[] = "Hello";
Char S2[] = "Hi";

for(i=0, S1[i]!='\0' || S2[i]!='\0', i++)
{
  if (S1[i] != S2[i])
    {
      flag = 1;
      break;
    }
}
if (flag==1)
  {
    printf ("strings are not same");
  }
```

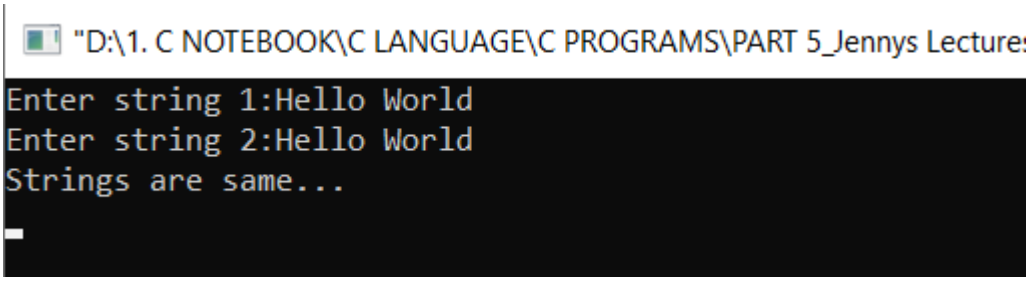
* For string comparison, in any loop we give condition set with a character != '\0', NULL character because compiler will understand a string having space __ & i.e Hai_Hello and Hai\0 both while space & null we be compared as same characters; so we get wrong and hence if we specify a '\0' character for comparison, compiler will understand.

CODE 1:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define N 50
/** 1 - String compare without using strcmp() **/
int main()
{
int i,flag=0;
char s1[N];
char s2[N];
printf("Enter string 1:");
gets(s1);
printf("Enter string 2:");
gets(s2);
for(i=0;s1[i]!='\0' || s2[i]!='\0';i++)
{
 if(s1[i]==s2[i])
  continue;
 flag=1;
 break;
}
if(flag==1)
 printf("Strings are not same...\n");
else
 printf("Strings are same...\n");
getch();
}
```


"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAM
Enter string 1:Hai
Enter string 2:Hai Hello
Strings are not same...


"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS
Enter string 1:Hai Hello
Enter string 2:Hai
Strings are not same...


"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lecture
Enter string 1:Hello World
Enter string 2:Hello World
Strings are same...

SAME CODE 1:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define N 50
/** 1 - String compare without using strcmp() **/
int main()
{
int i,flag=0;
char s1[N];
char s2[N];
printf("Enter string 1:");
gets(s1);
printf("Enter string 2:");
gets(s2);
for(i=0;s1[i]!='\0' || s2[i]!='\0';i++)
{
    if(s1[i]!=s2[i])
     {
      flag=1;
      break;
     }
 }
if(flag==1)
 printf("Strings are not same...\n");
else
 printf("Strings are same...\n");
getch();
}
```

CODE 2:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/** 2 - String compare using strcmp() **/
int main()
{
int value;
char s1[30];
char s2[30];
printf("Enter string 1:");
gets(s1);
printf("Enter string 2:");
gets(s2);
value=strcmp(s1,s2);
if(value==0)
printf("Strings are same...\n");
else
 printf("Strings are not same....\n");
printf("Value:%d",value);

getch();
}
```