

Strings

→ It is array of char. or a collection of char.

Character

It represents single letter within the "single" codes. eg: 'a', 'e', 'i', 'o', 'u'.

String: a collection of char. with in double codes.
eg "Hello".

→ If we want to include a double quote in the string to be printed, then we ~~can~~ may use it with a ~~back~~ back slash as shown below.

for eg:
printf ("I well done!")

result: "well done!"

(or)

printf ("well done!"); result: well done

The common operations performed on char. strings:

1. Declaring & initializing String Variables
2. Reading Strings from terminal
3. Writing Strings from terminal
4. Combining Strings together
5. Comparing Strings for equality
6. Copying one String to another

Declaring & Initializing String Variables:

Declaring:

It allows to represent strings as char array.

The general form of declaration of String Variable is;

`char String-name [Size of the String];`

The size delimites the no. of char in the String-name.

`char City [10];`

`char Name [30];`

When the compiler assigns a char string to a ~~for~~ a char array, it automatically supplies a null char (`"\0"`) at the end of the string.

`char City [9] = "NEW YORK";`

`char City [9] = {'N', 'E', 'W', '\0', 'Y', 'O', 'R', 'K', '\0'};`

Printing String for Terminal:

→ using scanf Function.

→ %s format specification to read in the string of char.

→ %s - used for accessing the string till it encountered `"\0"`.

Ex: char address[];
scanf("%s", address);

ex. pgm:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char str1[] = { 'H', 'E', 'L', 'L', 'O', '\0' };
    char str2[] = "Hello";
    clrscr();
    printf("%s\n", str1);
    printf("%s", str2);
    getch();
}
```

o/p: Hello
Hello

String operation:

→ a list of predefined library fun. in the header file <string.h> to carry out string manipulation.

- Important fun. of are,
1. strlen()
 2. strcpy()
 3. strcat()
 4. strcmp()

```
char str1[] = "Hello";
clrscr();
printf("%s", str1);
printf("%s", str1);
getch();
```

String length - strlen():

→ The fun. returns the num of char. in the string.

$n = \text{strlen}(\text{String})$.

where

n - is Integer

String - a string variables.

strcpy(): used for copying the source string into destination string.

(one string to another string)

strcpy (String1, String2)

strcmp(): used for comparing the string.

strcmp (String1, String2)

→ The two string are equal the value of the fun. will be zero. If the String1 is greater than String2 the value will be positive else negative.

strcat(): (string concatenating).

used for combining 2 strings.

strcat (String1, String2):

→ above fun. merges the contents of String2 with the String1.

→ When this fun. executed the contents of the Str2 are joined with String1, and the char '\0' is placed at the end of the new string String1.

- str 2 remains unchanged.
- The size of string 1 have enough place to keep the new string.

program:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
```

```
1 // string copy 1) str copy 2) strcpy
char str1[20] = "welcome";
char str2[10];
clrscr();
strcpy(str2, str1);
printf("%s", str2);
getch();
```

```
strcpy(str2, str1, 3);
str2[3] = '\0';
printf("%s", str2);
getch();
```

String Concatenating:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
```

```
void main():
```

```
1 // string concatenating 1) strcat 2) strncat
```

```
char str1[20] = "Hi guys";
char str2[10] = "welcome";
clrscr();
```

```

Strcat (str1, str2);
printf ("%s", str2);
getch();
}

```

O/p:

Hi guys, welcome

adder op
strcpy(str1,
str2[3])
↓
O/p
Hi guys, welcome

O/p:

Hi guys welcome.

String compare:

```
#include <stdio.h>
```

```
#include <conio.h>
```

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

```
void main()
```

```
{
```

1) String compare 2) strcmp

```
char str1[20] = "null";
```

```
char str2[10] = "null";
```

clrscr();

```
printf ("%d", strcmp(str1, str2));
```

```
getch();
```

O/p:

0

Step 2: (else condition)

```
n = strcmp(str1, str2);
```

```
if (n == 0)
```

```
printf ("Both strings are equal");
```

```
else if (n > 0)
```

```
printf ("String 1 is greater");
```

```
else printf ("String 2 is greater");
```

```

char str1[20] = "null";
char str2[10] = "male";
clrscr();

```

Value change as per

getch();

o/p: string is greater.

y

strcmp: a na strcmp (string 1, string 2)

o/p: Both string are equal

7	8	2	4	3	6	1	9
17	16	15	14	13	12	11	10

