

C Strings

1. Objective.....	2
2. Introduction.....	2
3. String Declaration & Initialization.....	2
4. C Built-in String Function	5
5. Questions/Practice	13

1. Objective

- What is a string?
- How do you declare and initialize a string?
- How can you use a string?
- Manipulating Strings in C
- String Examples
- String Practice

2. Introduction

- Sequence of zero or more characters, terminated by NUL (literally, the integer value 0)
- Every string is terminated by NUL and NUL is not part of the string.

3. String Declaration & Initialization

- A string in C is nothing but an array of type **char**
- Two ways to declare a variable that will hold a string of characters:
 - Using arrays:

```
char mystr[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

- Using a string of characters:

```
char mystr [] = "Hello";
```

H	E	L	L	O	\0
---	---	---	---	---	----

- Printing Strings:
 - Can print an entire string using printf and %s format specification
 - Can print individual elements of a string by indexing and using %c format specification

- Example:

```
//gcc 5.4.0
#include <stdio.h>
int main(void)
{
    char mystr1[] = "Fox Music!";
    char mystr2[10] = "Fox Music!";
    char mystr3[4];

    printf("mystr1 = %s\n", mystr1);

    for (int i=0; i<sizeof(mystr1)/sizeof(char); ++i)
        printf("mystr[%d] = %c\n", i, mystr1[i]);
}
```

```
printf("-----\n");

for (int i=0; i<sizeof(mystr2)/sizeof(char); ++i)
    printf("mystr2[%d] = %c\n", i, mystr2[i]);

printf("-----\n");

mystr3[0] = 'P';
mystr3[1] = 'r';
mystr3[2] = 'o';
mystr3[3] = 'g';

printf("mystr3 = %s**\n", mystr3);

return (0);
}
```

4. C Built-in String Function

- The library **string.h** contains prototypes of many useful functions:

Strcpy(dest, src)	copies src string into dest string.
strncpy	copies a certain amount of characters from one string to another
strcat (str1, str2);	str2 is concatenated at the end of str1.
strncat (str1, str2, n);	First n characters of str2 is concatenated at the end of str1.
strlen(str1)	Calculates the length of string
strcmp(str1, str2)	Compares two string

- Strcpy(dest, src)
 - It copies one string into another string.

Strcpy(dest, src)

- Example:

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

```
int main()
```

```
{
```

```
    char src[40];
```

```
    char dest[100];
```

```
    strcpy(src, "Computer Science");
```

```
    strcpy(dest, src);
```

```
    printf("String to be copied: %s\n", src);
```

```
    printf("Final copied string : %s\n", dest);
```

```
    return(0);
```

```
}
```

- strncpy ()

- strncpy() function copies portion of contents of one string into another string.

- Syntax:

```
strncpy (dest, src, size_t num );
```

- if dest string length is less than src string, entire src string value won't be copied into dest string.
- Example:

```

#include <stdio.h>
#include <string.h>
int main( )
{
    char src[25] = "C Programming Language" ;
    char dest[40]= "" ;
    char anotherdest[10] = "";

    printf ( "\nsource string = %s", src) ;
    printf ( "\ndest string = %s", dest) ;
    strncpy ( dest, src, 15 ) ;
    printf ( "\ndest string after strcpy( ) = %s", dest) ;

    //The numbe of character to be copied is greater
    that length of dest.
    strncpy ( anotherdest, src, 15 ) ;
    //Note the Memory map that is printed in this
    version of the compiler
    printf ( "\ndest string after strcpy( ) = %s",
    anotherdest) ;
    return 0;
}

```

- How to find the length of a string?
 - Two ways:
 - Using the built-in strlen() function
 - Using the special symbol that indicates the end of the string.
 - The built-in function, strlen(), does not include the NUL:

```
char mystr3[2];  
printf("The length of mystr3 is: %d\n", strlen(mystr3));
```

```
//gcc 5.4.0  
#include <stdio.h>
```

```
void main(void)  
{  
    char mystr1[] = "Fox Music!";  
    char mystr2[10] = "Fox Music!";  
    char mystr3[2];  
  
    printf("The length of mystr3 is: %d\n",  
strlen(mystr3));
```

```
    //strcoll: compares two strings in accordance to the  
current locale (Language: Eng., French, etc.)  
    if(strcoll(mystr1, mystr2) == 0)  
        printf("equal\n");  
    if(strcoll(mystr1, mystr3) == 0)  
        printf("equal\n");
```



```

else
    printf("Not equal\n");

strcpy(mystr3, mystr1);
printf("The length of mystr1 is: %d\n",
strlen(mystr1));
printf("The length of mystr2 is: %d\n",
strlen(mystr2));
printf("The length of mystr3 is: %d\n",
strlen(mystr3));

printf("mystr3=%s\n", mystr3);
for (int i=0; i<strlen(mystr3); ++i)
    printf("mystr3[%d] = %c\n", i, mystr3[i]);
}

```

○ Example:

```

//gcc 5.4.0

#include <stdio.h>
int main()
{
    char s[1000] = "Hello World!";
    int i;

    for(i = 0; s[i] != '\0'; ++i);

    printf("Length of string: %d\n", i);
    printf("Length of string using strlen(): %d", strlen(s));
}

```

```
    return 0;
}
```

- `strcat()`:
 - It is used to concatenate two strings
 - Example:

```
//gcc 5.4.0
```

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

```
int main(void)
```

```
{
```

```
    char source[ ] = "C Programming" ;
```

```
    char destination[20]= "" ;
```

```
    printf ( "\nsource string = %s", source ) ;
```

```
    printf ( "\ntarget string = %s", destination) ;
```

```
    strncpy ( destination, source, 6 ) ;
```

```
    printf ( "\ntarget string after strcpy( ) = %s",  
destination) ;
```

```
    return 0;
```

```
}
```

- strcmp():
 - It compares two strings character by character
 - It returns the following:

0:	if both strings are identical (equal)
Negative:	if the ASCII value of first unmatched character is less than second.
positive integer:	if the ASCII value of first unmatched character is greater than second.

- Example:

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

```
int main()
{
    char src[40];
    char dest[100];
```

```
    strcpy(src, "Computer Science");
```

```
strcpy(dest, src);

printf("String to be copied: %s\n", src);
printf("Final copied string : %s\n", dest);
printf("Compare the two strings:%d\n", strcmp(src,
dest));
strcpy(src, "Computer Science Department");
printf("Compare the two strings:%d\n", strcmp(src,
dest));
return(0);
}
```

5. Questions/Practice

- Write a C program that decodes the following common tweet abbreviations:

LOL or lol or Lol → Laughing Out Loud

BFN or bnf or Bnf → Bye For Now

FTW or ftw or Ftw → For The Win

IRL or irl or Irl → In Real Life

Note: Ask the user for the abbreviation.

- Write a C program that implements that stores roster and rating information for a soccer team:
 - A player's jersey number is between 1 and 99
 - A player's rating is between 1 and 9

Create a structure to store a player information and an array to store the team information.

Note: You can use the following a random number between 1-9 and 1-99

```
printf("%d\n",rand() % 9 + 1);
```

```
printf("%d\n",rand() % 99 + 1);
```