***Chapter 8 - Strings***

A string is a 1-d character array terminated by a null(‘\0’) => {this is null character}

The null character is used to denote string termination, characters are stored in contiguous memory locations.

**Initializing Strings**

Since string is an array of characters, it can be initialized as follows:

char s[]={‘H’,’A’,’R’,’R’,’Y’,’\0’}

Copy

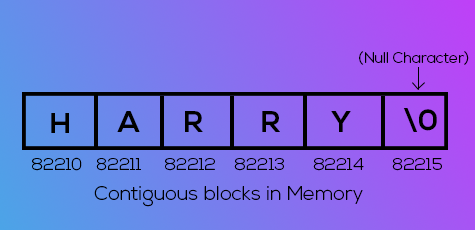
There is another shortcut for initializing strings in c language:

char s[]=”HARRY”; => In this case C adds a null character automatically.

Copy

**Strings in memory**

A string is sorted just like an array in the memory as shown below



**Quick Quiz:**Create a string using " " and print its content using a loop.

**Printing Strings**

A string can be printed character by character using printf and %c.

But there is another convenient way to print strings in C.

char st[] = ”HARRY”;

printf(“%s”,st); => prints the entire string

Copy

**Taking string input from the user**

We can use %s with scanf to take string input from the user:

char st[50];

scanf(“%s”,&st);

Copy

scanf automatically adds the null character when the enter key is pressed.

**Note:**

1. The string should be short enough to fit into the array.
2. scanf cannot be used to input multi-word strings with spaces.

**gets() and puts()**

gets() is a function that can be used to receive a multi-word string.

char st[30];

gets(st); => the entered string is stored in st!

Copy

Multiple gets() calls will be needed for multiple strings.

Likewise, puts can be used to output a string.

puts(st);   =>Prints the string and places the cursor on the next line

**Declaring a string using pointers**

We can declare strings using pointers

char \*ptr= ”Harry”;

Copy

This tells the compilers to store the string in the memory and the assigned address is stored in a char pointer.

**Note:**

1. Once a string is defined using char st[]= ”harry”, it cannot be initialized to something else.
2. A string defined using pointers can be reinitialized.   => ptr=”rohan”;

**Standard library functions for Strings**

C provides a set of standard library functions for strings manipulation.

Some of the most commonly used string functions are:

**strlen() -**This function is used to count the number of characters in the string excluding the null ('\0') character.

int length=strlen(st);

Copy

These functions are declared under <string.h> header file.

**strcpy() -**This function is used to copy the content of second string into first string passed to it.

char source[ ]= ”Harry”;

char target[30];

strcpy(target,source); => target now contains “Harry”

Copy

Target string should have enough capacity to store the source string.

**strcat() -**This function is used to concatenate two strings

char s1[11]= ”Hello”;

char s2[ ]= ”Harry”;

strcat(s1,s2); => s1 now contains “Hello Harry” <No space in between>

Copy

**strcmp() -**This function is used to compare two strings. It returns: 0 if strings are equal

Negetive value if first strings mismatching character's ASCII value is not greater than second string's corresponding mismatching character. It returns positive values otherwise.

strcmp(“For”, “Joke”); => positive value

strcmp(“Joke”, “For”); => Negative value

Copy

Chapter 8 - Practice Set

1. Which of the following is used to appropriately read a multi-word string-

* Gets()
* Puts()
* Printf()
* Scanf()

1. Write a program to take a string as an input from the user using %c and %s. Confirm that the strings are equal.
2. Write your own version of strlen function from <string.h>
3. Write a function slice() to slice a string. It should change the original string such that it is now the sliced strings. Take m and n as the start and ending position for slice.
4. Write your own version of strcpy function from <string.h>
5. Write a program to encrypt a string by adding 1 to the ASCII value of its characters.
6. Write a program to decrypt the string encrypted using the encrypt function in problem 6.
7. Write a program to count the occurrence of a given character in a string.
8. Write a program to check whether a given character is present in a string or not.