Chapter 2 : Indexed variables

Part III : Strings in C

Definition:

A string in C is an array of characters terminated by a null character "\0". Simply, a string contents are the array characters plus an extra character that is called null character.

Declaration of a string:

A string it can be declared in the same way as an array of characters:

char string_name[size];

Example:

char S[4]= "abc";

a	b	c	\0
S[0]	S[1]	S[2]	S[3]

Initialization of a string:

There are two ways to initialize a string:

1- Direct initialization:

Example:

Program	*Output*

#include <stdio.h>
 	
int main() {
char letter[]="Good to see you";
int i,
length=sizeof(letter)/sizeof(letter[0]);
printf("String length=%d\n",length);
printf("String=%s\n", letter);
printf("String elements:\n");
for (i = 0; i < length; ++i)
{
printf("%c\n", letter[i]);
}
return 0;
}
	String length=16
	String=Good to see you
	String elements:
	G
	o
	o
	d
	
	t
	o
	
	s
	e
	e
	
	y
	o
	u
	.

It is worth note that the “space” is counted and the null character is printed as a point (“.”).

2- Initiaialization during declaration as an aarray:

Example:

Program	*Output*

#include <stdio.h>
 	
int main() {
char letter[]={'G','o','o','d',' ','t','o',' ','s','e','e',' ','y','o','u','\0'};
printf("String=%s\n", letter);
 	
return 0;
}
	String=Good to see you

String functions:

The most functions used to deal with strings are:

String Library Functions

#include <string.h>

Name	Description
strlen	return the length of string not counting \0
strcopy	copies string from source to dest
strncopy	copies n chars from source to dest
strcat	appends string from source to end of dest
strncat	appends n chars from source to end of dest
strcmp	compares two strings alphabetically
strncmp	compares the first n chars of two strings
strstr	finds a string inside another
strtok	breaks string into tokens using delimiters

Dr. Sadık Eşmelioğlu CENG 114 2

Figure1: C String functions
Image source: https://onlinestores.best2024.ru/category?name=c%20%20%20string%20library%20functions

Some operations on strings:

1- Modify strings:

Program	*Output*

#include <stdio.h>
 	
int main() {
char message[11]="My friend";
message[0]='H';
message[1]='i';
message[9]='!';
printf("%s", message);
 	
return 0;
}
	Hi friend!

2- Program to find number of words in a string

Program	*Output*

#include <stdio.h>
#include <string.h>
 	
void main()
{
char s[200];
int count = 0, i;
 	
printf("Enter the string:\n");
scanf("%[^\n]s", s);
for (i = 0;s[i] != '\0';i++)
{
if (s[i] == ' ' && s[i+1] != ' ')
count++;
}
printf("Number of words in given string are: %d\n", count + 1);
}
	Enter the string:
	
	Don't apologize for not understanding. If you stop asking questions then you effectively kill your desire to know the unknown
	
	Number of words in given string are: 20