

STRINGS

What are Strings

- Character arrays may also called strings.
- Character arrays or strings are used by programming languages to manipulate text such as words and sentences.
- A string constant is a one-dimensional array of characters terminated by a null (`'\0'`).

Ex:

```
char name[ ] = { 'H', 'A', 'E', 'S', 'L', 'E', 'R', '\0' } ;
```

- Each character in the array occupies one byte of memory and the last character is always `'\0'`.
- `'\0'` is called null character.
- Note that `'\0'` and `'0'` are not same. ASCII value of `'\0'` is 0, whereas ASCII value of `'0'` is 48.

What are Strings

- Note that the elements of the character array are stored in contiguous memory locations.
- The terminating null ('\0') is important, because it is the only way the functions that work with a string can know where the string ends. In fact, a string not terminated by a '\0' is not really a string, but merely a collection of characters.

H	A	E	S	L	E	R	\0
65518	65519	65520	65521	65522	65523	65524	65525

Initializing strings

- ⦿ `char c[] = "abcd";`
- ⦿ `char c[50] = "abcd";`
- ⦿ `char c[] = {'a', 'b', 'c', 'd', '\0'};`
- ⦿ `char c[5] = {'a', 'b', 'c', 'd', '\0'};`

Example : scanf() to read a string

```
#include <stdio.h>

void main()
{
char a[20];
printf("Enter name: ");
scanf("%s", a);
printf("Your name is %s", a);
}
```

Example - String

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

```
void main()
```

```
{
```

```
clrscr();
```

```
char name[]="Viva Technologies";
```

```
int i;
```

```
for(i=0;i<16;i++)
```

```
{
```

```
printf("%c",name[i]);
```

```
}
```

```
getch(); }
```

Example - String

```
#include<stdio.h>
#include<conio.h>
void main()
{
char name[25];
clrscr();
printf("enter your name;\n");
scanf("%s",name);
printf("your name is -:%s\n",name);
getch(); }
```

Example : fgets() and puts()

```
#include <stdio.h>
void main()
{
char name[30];
printf("Enter name: ");
fgets(name, sizeof(name), stdin); // read string
printf("Name: ");
puts(name); // display string
getch();
}
```

You can use the fgets() function to read a line of string. And, you can use puts() to display the string.

Example - String

```
void main()
{
char name[30];
clrscr();
printf("enter your name:");
scanf("%s",name);
gets(name);
printf("hello your name is :%s\n",name);
puts("hello ");
puts("name");
getch();
}
```

Passing Strings to Functions

Passing Strings to Functions

```
#include <stdio.h>
void displayString(char str[]);
void main()
{
    char str[50];
    printf("Enter string: ");
    fgets(str, sizeof(str), stdin);
    displayString(str); // Passing string to a function.
}
```

```
void displayString(char str[])
{
    printf("String Output: ");
    puts(str);
}
```

Strings and Pointers

```
#include <stdio.h>

void main
{
char name[] = "Harry";
printf("%c", *name); // Output: H
printf("%c", *(name+1)); // Output: a
printf("%c", *(name+4)); // Output: o
}
```

Standard Library String Functions

Operations on Strings

- Reading and displaying strings.
- Copying one string to another.
- Combining strings.
- Comparing strings.
- Extraction of a portion of a strings.
- checking string for equality.

- strcpy : Copies a string into another.
- strncpy : Copies first n characters of one string into another.
- strcmp : Compares two strings.
- strncmp : Compares first n characters of two strings.
- strcmpi : Compares two strings without regard to case ("i" denotes that this function ignores case).
- stricmp : Compares two strings without regard to case (identical to strcmpi).
- strnicmp : Compares first n characters of two strings without regard to case.

FINDING LENGTH OF A STRING - strlen() function

```
#include <stdio.h>
#include <string.h>
void main()
{ char a[20]="Program";
char b[20]={'P','r','o','g','r','a','m','\0'};
char c[20]="Viva Technologies";
printf("Length of string a = %d \n",strlen(a));
printf("Length of string b = %d \n",strlen(b));
printf("Length of string c = %d \n",strlen(c));
getch();
}
```


COPY STRING - strcpy()

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

void main()
{
    char str1[20] = "Viva Technologies";
    char str2[20];
    strcpy(str2, str1); // copying str1 to str2
    puts(str1); // Printing Viva Technologies
    puts(str2); // Printing Viva Technologies
    getch() }
```

STRING COMPARISON- strcmp() function

```
void main()
{
char str1[] = "A", str2[] = "B";
int result;
result = strcmp(str1, str2); // comparing strings str1 and str2
printf("strcmp(str1, str2) = %d\n", result);
}
```

ASCII - Binary Character Table

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
A	065	01000001	N	078	01001110
B	066	01000010	O	079	01001111
C	067	01000011	P	080	01010000
D	068	01000100	Q	081	01010001
E	069	01000101	R	082	01010010
F	070	01000110	S	083	01010011
G	071	01000111	T	084	01010100
H	072	01001000	U	085	01010101
I	073	01001001	V	086	01010110
J	074	01001010	W	087	01010111
K	075	01001011	X	088	01011000
L	076	01001100	Y	089	01011001
M	077	01001101	Z	090	01011010

ASCII - Binary Character Table

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	n	110	01101110
b	098	01100010	o	111	01101111
c	099	01100011	p	112	01110000
d	100	01100100	q	113	01110001
e	101	01100101	r	114	01110010
f	102	01100110	s	115	01110011
g	103	01100111	t	116	01110100
h	104	01101000	u	117	01110101
i	105	01101001	v	118	01110110
j	106	01101010	w	119	01110111
k	107	01101011	x	120	01111000
l	108	01101100	y	121	01111001
m	109	01101101	z	122	01111010

STRING COMPARISON- strcmp() function

```
void main()
{
char str1[] = "Berry", str2[] = "Ferry", str3[] = "Jerry";
int result;
result = strcmp(str1, str2); // comparing strings str1 and str2
printf("strcmp(str1, str2) = %d\n", result);
result = strcmp(str1, str3); // comparing strings str1 and str3
printf("strcmp(str1, str3) = %d\n", result);
}
```

JOINING OR CONCATENATION - strcat() function

```
main()
```

```
{
```

```
char str1[] = "Viva ", str2[] = "Technologies";
```

```
puts(str1);
```

```
puts(str2);
```

```
strcat(str1, str2); // concatenates str1 and str2
```

```
}
```

```
// the result string is stored in str1.
```

JOINING OR CONCATENATION - strcat() function

```
main()
```

```
{
```

```
char name[] = "Viva", surname [ ] = "Technologies";
```

```
strcat(name, surname); // concatenates str1 and str2
```

```
puts(name);
```

```
strcat(surname, name); // concatenates str2 and str1
```

```
puts(surname);
```

```
}
```

THANKS