STRING IN 'C'

→ A string is a sequence of characters that is treated as a single data item.

→ for example: "My name is deepak"

Declaration & Intialization of String

→ 'C' represent string as a character array.

→ The general form of declaration of a string is:

```
char string_name[size];
```

Example:-

```
char city[10];
char name[30];
```

→ When Compiler assign a character string to a character array, it automatically supplies a null character ('\0') at the end of the string.

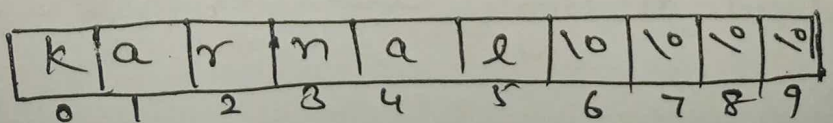
→ The character array is intialized when they are declared as:

```
char city[9] = "India"
```

or

```
char city[9] = {'K', 'a', 'r', 'n', 'a', 'l', '\0', '\0', '\0', '\0'};
```

→ The storage will look like:



How to read string?

1) Using Scanf function :-

→ Scanf can be used with %s formate specification to read in a string.

→ Example:
```
char address[10];
scanf("%s", address);
```

→ The problem with scanf function is that it terminates its input on the first white space it finds.

→ for Example: New york is read only as "new", since blank space after the word 'new' will terminate the reading of string.

Example:-
```
#include<stdio.h>
#include<string.h>
main()
{
char name[10];
printf("Enter text");
scanf("%s", name);
}
```

output :→ Enter text: Deepak.

2) Using getchar & gets function :-

→ getchar() read a single character from the termi

Example:- char ch;
ch = getchar();

* getchar function has no parameter.

→ Another more effective methode of reading a strin of text containing whitespace is to use the librar function gets available in the <stdio.h> header file.

→ for example:- gets(str);

→ 'str' is a string variable declared properly.

Example:- char line[80];
gets(line);
printf("%s", line);

```
#include<stdio.h>
#include<conio.h>
main()
{
char a[20];
printf("Enter the string");
gets(a);
}
```

output: enter the string: Deepak.

WRITING STRING TO SCREEN.

1) using printf function:- printf function with %s form to print strings to the screen.

Example:- printf("%s", name);

2 Using putchar & puts formate :-

→ putchar function is used output the values of character variable.

→ Syntax:- char ch='A'
putchar(ch);

Example:- char name[6] = "Paris"
for(i=0; i<5; i++)
putchar(name[i]);

→ Another more effective way of printing string values is to use the function puts declared in the header file <stdio.h>.

→ Syntax:- puts(str);

Example:- char line[80];
gets(line);
puts(line);

STRING HANDLING FUNCTION :-

→ following are the most commonly used function in string.

1) Strcat()
2) Strcmp()
3) Strcpy()
4) Strlen()

1) strcat() :→ This function concatenates the source string at the end of the target string.

Example:-

```
#include <stdio.h>
#include <string.h>
int main()
{
    char source[] = "Annu";
    char target[] = "sharma";
    strcat (target, source);
    printf("After Concatenation %os", target);
    return 0;
}
```

2) strcmp() :→ This is a function which compare two strings to find out whether they are same or different.

→ If two strings are identical it returns 0.
→ If they are not, it returns the numeric difference between the ASCII values of the first non-matching pair of characters.

Example:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char string1[] = "Deepak";
    char string2[] = "singla";
    int i, j, k;
    i = strcmp(string1, "Deepak");
    j = strcmp(string2, "singla");
    printf("%d %d", i, j);
    return 0;
}
```

3 Strcpy() :- This function copies the content of one string into another.

→ The base address of source strings and destination strings should be supplied to this function.

```
#include <stdio.h>
#include <string.h>
main()
{
    char source[] = "Deepak";
    char target[] = "Singla";
    strcpy(target, source);
    printf("After copy %s", target);
    return 0;
}
```

Output:- After Copy Deepak

4 Strlen() :- This function counts the number of character present in a string.

Example:-

```
#include<stdio.h>
#include<string.h>
int main()
{
    char arr[] = "Deepak"
    int length;
    length = strlen(arr);
    printf("length of string is %d", length);
    return 0;
}
```

Output:- length of string is 6.