String Library Functions

strlen – find the length of the string.

Syntax :int variable_name= strlen(string_name);

strcpy

- copies the contents of one string into another.
- base addresses of the source and target strings should be supplied to this function.
- main( )

```
main( )
 {
        char source[20] = "Sayonara" ;
        char target[20] ;
        strcpy ( target, source ) ;
        printf ( "\nsource string = %s", source ) ;
        printf ( "\ntarget string = %s", target ) ;
 }
```

Output:

```
source string = Sayonara
 target string = Sayonara
```

- On supplying the base addresses, strcpy( ) goes on copying the characters in source string into the target string until it encounter the end of source string (‘\0’)
- without using strcpy

```
main()
{
        char source[20] = "Sayonara“, target[20] ;
        char *s,*t;
        s=source;
        t=target;
        while ( *s != '\0’ )
        {
                *t = *s ;
                 s++ ;
                 t++;
        }
        *t=‘\0’;
}
```

strcat

- concatenates the source string at the end of the target string.
- For eg, “Bombay” and “Nagpur” on concatenation would result into a string “BombayNagpur”.
-

```
main( )
{
          char source[30] = "Folks!" ;
         char target[30] = "Hello" ;
         strcat ( target, source ) ;
         printf(“source string =“);
         puts(source);
         printf(“target string =“);
          puts(target);
 }
```

And here is the output...

```
 source string = Folks!
target string = HelloFolks!
```

Without using strcat

```
main()
{
        char source[30] = "Folks!" ,target[30] = "Hello" ;
        int i ;
        for(i=0;target[i]!=‘\0’;i++);
        for(j=0;source[j]!=‘\0’;i++,j++)
        {
                target[i]=source[j];
        }
        target[i]=‘\0’;
        printf ( "\nsource string = %s", source ) ;
         printf ( "\ntarget string = %s", target ) ;
}
```

And here is the output...

```
 source string = Folks!
target string = HelloFolks!
```

strcmp

- compares two strings to find out whether they are same or different.

- The two strings are compared character by character until there is a mismatch or end of one of the strings is reached, whichever occurs first.

- If the two strings are identical, strcmp( ) returns a value zero. If they’re not, it returns the numeric difference between the ASCII values of the first non-matching pairs of characters.

- main( )

```
 {
        char string1[ ] = "Jerry" ,string2[ ] = "Ferry" ;
         int  j;
        j=strcmp ( string1, string2 ) ;
        if(j==0)
                printf(“Strings are equal”);
        else
                printf(“Strimgs are not equal”);
 }
```

❑Without using strcmp

```
#include<stdio.h>
main()
{

        char string1[ ] = "Jerry" ,string2[ ] = "Ferry" ;

         int  i=0;

        while((string1[i]==string2[i]) &&(((string1[i] != ‘\0’)||(string2[i]!=‘\0’)))

        {

                i++;

        }

        if(string1[i] !=string2[i])

                printf(“Strings are not equal”);

        else

                printf(“Strings are equal”);

}
```