String

String is an array of characters. It is used to store non-numerical values like name, address, subject name etc. the last character of string is always be a NULL ('\0') character which is used to terminate the string. String is denoted by double quote (""). **char** data type is used to define the string. **%s** is a format specifier for string literal. In higher languages you will be able to use *string class* for string operation but in c language you need to create an array of char to define string.

String Declaration

an array of characters is a string, so we need to create the array with char data type.

Syntax :

char string[size];

Example :

char name[10];

There always needs one extra character in size for the NULL ('\0'). however this thing only matters in compiler time initialization. In run time even approx size will be work as accurate value.

String initialization

1. compile time initialization

In compile time, programmer could pass all characters as a single value in double quote like,

Syntax :

char string[size] = "string literal";

Example :

char name[10] = "Computer";

or

all characters with personal single quote with separated by semicolon in curly brackets.

Syntax :

char string[size] = {'s', 't', 'r', 'i', 'n', 'g', ' ', 'l', 'i', 't', 'e', 'r', 'a', 'l', '\0'} ;

Example :

char name[10] = {'C', 'o', 'm', 'p', 'u', 't', 'e', 'r', '\0'};

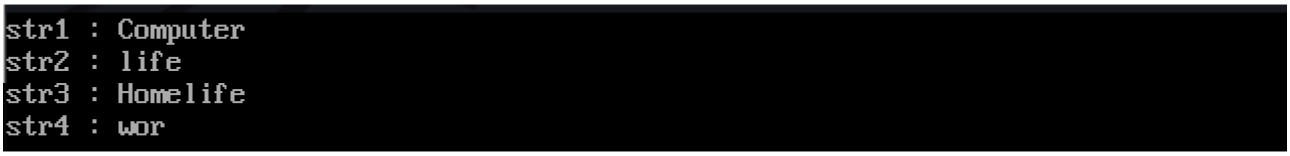
As you the last character is **'\0'**. which is used to terminate the string, if you don't provide the **NULL** (**'\0'**) then compiler will read some garbage characters.[NOTE : it(NULL '\0') is necessary when you initialize the string with single quote.]

Size is optional is compile time initialization, if you mention the size it must be equals or greater than number of characters.

Program 1: Write program to display a predefine string in console.

```
#include<stdio.h>
#include<conio.h>
void main( ){
   char str1[] = "Computer"; // also work with size.
   char str2[5]= {'l','i','f','e'}; // 1 extra size for NULL
   char str3[4] = {'H','o','m','e'}; // gives garbage string due to absent of NULL position.
  char str4[10] = {'w','o','r','\0','l','d'}; //display wor, l,d will terminated by \0
   clrscr();
          printf("str1 : %s\n",str1);
          printf("str2 : %s\n",str2);
          printf("str3 : %s\n",str3);
          printf("str4 : %s\n",str4);
   getch();
}
```

OUTPUT:



```
str1 : Computer
str2 : life
str3 : Homelife
str4 : wor
```

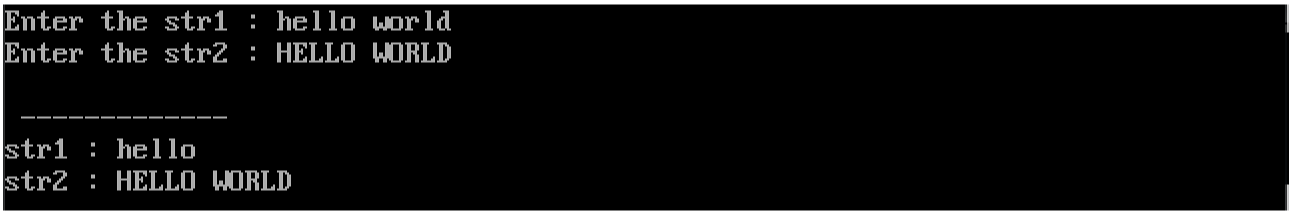
2. run time initialization

In run time you no need to worry about the NULL(‘\0’) character. You can use either scanf() or gets() for the initialization. But there is a big difference between them. scanf() can initialize the only a single word to the string. However gets() can take a line with multiple words to the string. Lets see that by the example.

Program 2: Write program to take two strings from user.

```
#include<stdio.h>
#include<conio.h>
void main()
{
     char str1[100];
     char str2[100];
clrscr();
          printf("Enter the str1 : ");
          scanf("%s",str1); // no need of & due to string.

           flushall(); // to clean the console inputs

          printf("Enter the str2 : ");
          gets(str2);

          printf("\n ------------- \n");

          printf("str1 : %s\n",str1);
          printf("str2 : %s\n",str2);
getch();
}
```

OUTPUT:



```
Enter the str1 : hello world
Enter the str2 : HELLO WORLD

 -------------
str1 : hello
str2 : HELLO WORLD
```

See str1 couldn’t able store more than one word. NOTE gets() is available in TURBO C++ compiler. In other getline() used.

String Operations

string operations helps you to manipulate the string literal. You can use either a function or build the logic for string operation. Its good to use in-build function but to build the logic you need to understand how the function worked. All function for string operation are available in **string.h** header file.

No.	Operation	Function	Syntax
1	Find length of str.	strlen()	int strlen(char [])
2	Comparing 2 strings.	strcmp()	int strcmp(char [], char [])
3	Copy one str to another.	strcpy()	void strcpy(char [],char [])
4	Concatenate two strings.	Strcat()	void strcat(char [],char [])
5	Convert str into uppercase.	strupr()	void strupr(char [])
6	Convert str into lowercase.	strlwr()	void strlwr(char [])
7	Finding a character in str.	strchr()	char * strchr(char [],char)
8	Finding sub str in the str.	strstr()	char * strstr(char [],char [])
9	Reverse the str.	strrev()	void strrev(char[])
10	Mask the str.	strset()	void strset(char [],char)

1. Calculate length of string.

i. With function : strlen() is used for this task. This function will return an integer value as a length of string.

Syntax :

int strlen(char []);

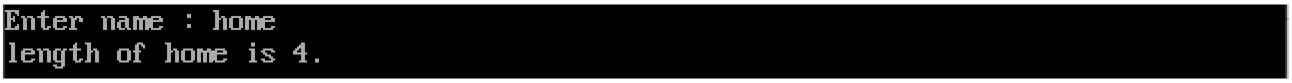
Example :

len = strlen(“Home”);

Program 3: Write program to find the length of string.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
   char name[100];
   int len;
clrscr();
          printf("Enter name : ");
          gets(name);
          len = strlen(name);
          printf("length of %s is %d.\n",name,len);
getch();
}
```

OUTPUT:



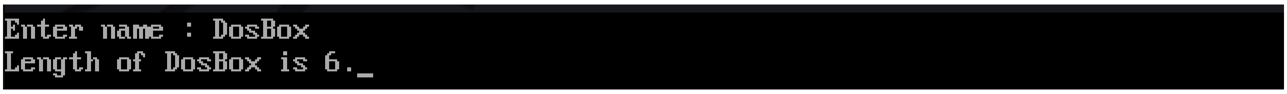
```
Enter name : home
length of home is 4.
```

ii. Without function. Iterate all characters & increment the counter variable to calculate the length.

Program 4: Write program to find the length of string without strlen().

```
#include<stdio.h>
#include<conio.h>
void main( )
{
  char name[100];
  int len=0,i;
clrscr( );
       printf("Enter name : ");
       gets(name);

       for(i=0; name[i]; i++){
            len++;   // increment till the NULL character.
       }
       printf("Length of %s is %d.",name,len);
getch( );
}
```

OUTPUT:



```
Enter name : DosBox
Length of DosBox is 6._
```

2. *Comparing 2 strings.*

I. With function: strcmp() is used for this task. strcmp() returns 0 if both string are same else subtraction of 1st mismatched characters.

Syntax :

int strcmp(char [],char []);

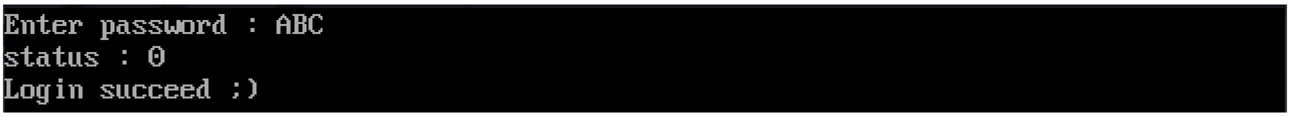
Example :

status = strcmp("Hello", "Bye");

Program 5: Write program to compare 2 strings.

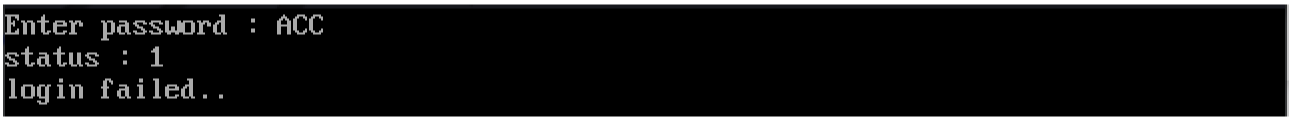
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main( )
{
          char password[10] = "ABC"; //default value for password
          char input[10];
          int status;
clrscr( );
          printf("Enter password : ");
          gets(input);

          status = strcmp(input,password);
          printf("status : %d\n",status);
          if(!status) // if status == 0
          {
            printf("Login succeed ;) \n");
          }
          else
          {
            printf("login failed.. \n");
          }
getch();
}
```

OUTPUT 1:



```
Enter password : ABC
status : 0
Login succeed ;)
```

OUTPUT 2:



```
Enter password : ACC
status : 1
login failed..
```

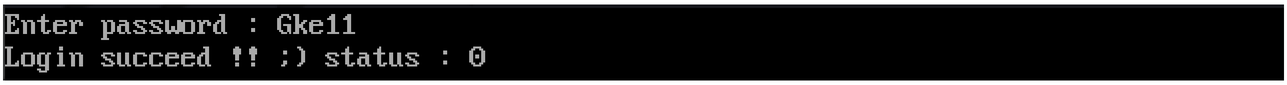
II. Without function in this manner, Programmer can check character by character of both string to compare. However some programmer First check the length of string & then check all corresponding characters. We'll take a loop until one string is not over one of them by for loop iteration. Here we'll learn another version of for loop too…

Program 6: Write program to compare 2 strings without strcmp().

```
#include<stdio.h>
#include<conio.h>
void main()
{
     char password[10] = "Gke11";
     char input[10];
     int status=0,i;
clrscr();
     printf("Enter password : ");
     gets(input);

     for(i=0;input[i];i++)
     {
          if(password[i]!=input[i])
          {
             status = password[i]-input[i]; //ASCII deference
             break;
          }
     }
     if(!status) // if status is 0.
          printf("Login succeed !! ;) status : %d\n",status);
```

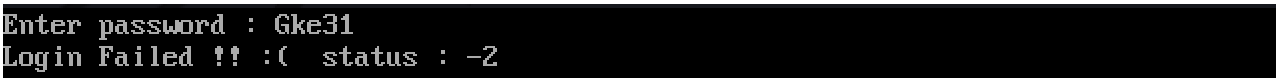
```
    else
        printf("Login Failed !! :(  status : %d\n",status);
getch();
}
```

OUTPUT 1:



```
Enter password : Gke11
Login succeed !! :) status : 0
```

OUTPUT 2:



```
Enter password : Gke31
Login Failed !! :(  status : -2
```

3. *Copy one String into another.*

I. With function : We have strcpy() to copy one string into another string.

Syntax :

void strcpy(char[],char[]);

Example :

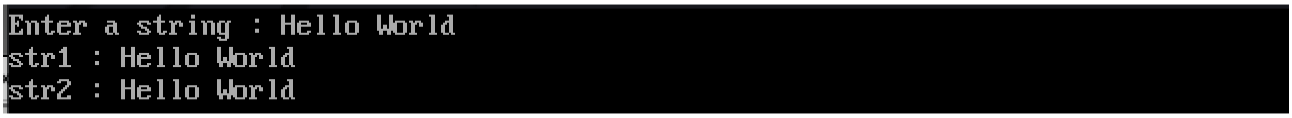
strcpy(str1,str2);

above example copy str1 into str2.

Program 7: Write program to copy one string into another using strcpy().

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char str1[100],str2[100];
clrscr();
        printf("Enter a string : ");
        gets(str1);
        strcpy(str2,str1); //str2 ← str1
        printf("str1 : %s\n",str1);
        printf("str2 : %s\n",str2);

getch();
}
```

OUTPUT:



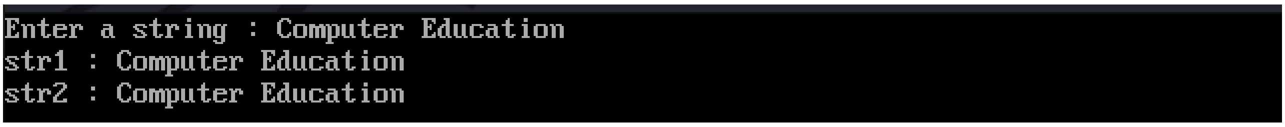
```
Enter a string : Hello World
str1 : Hello World
str2 : Hello World
```

II. Without function : We need to transfer one by one character to another string with help of loop & end of string a ‘\0’ character must be there for terminating string.

Program 8: Write a program to copy one string into another without function.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        char str1[100],str2[100];
        int i;
clrscr();
        printf("Enter a string : ");
        gets(str1);

        i=0;
        while(str1[i]) // str1[i] is not NULL(‘\0’)
        {
          str2[i]=str1[i]; //str2[i] ← str1[i]
          i++; //target next character
        }
        str2[i] = '\0';
        printf("str1 : %s\n",str1);
        printf("str2 : %s\n",str2);
getch();
}
```

OUTPUT:



```
Enter a string : Computer Education
str1 : Computer Education
str2 : Computer Education
```

4. *Concatenate two strings.*

I. With function : Concatenation means appending string at end of another string.strcat() helps us in this manner

Syntax:

void strcat(char [],char[])

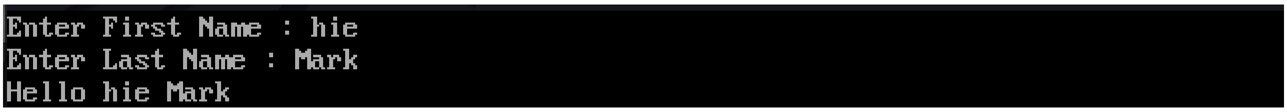
Example:

strcat(str1,str2)

Program 9: Write a program to join 2 strings with strcat().

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char str1[100],str2[10];
clrscr();
```

```
        printf("Enter First Name : ");
        scanf("%s",str1);

        printf("Enter Last Name : ");
        scanf("%s",str2);

        strcat(str1," "); // adding space : str1 = str1 + " "
        strcat(str1,str2); //adding str2 after space: str1 = str1+str2

        printf("Hello %s\n",str1);
getch();
}
OUTPUT:
```

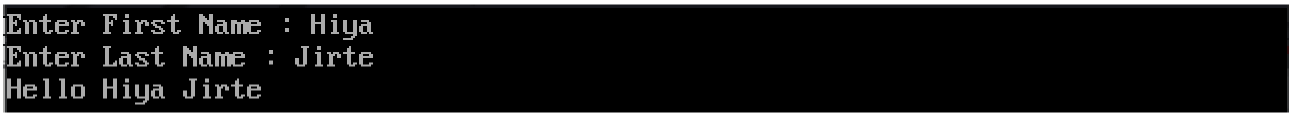


```
Enter First Name : hie
Enter Last Name : Mark
Hello hie Mark
```

II. Without function : Iterate all characters of string 1 till end & them start insert one by one character of string 2 into string 1.

Program 10: Write a program to merge(join) 2 strings.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        char str1[100],str2[100];
        int i,j;
clrscr();
        printf("Enter First Name : ");
        scanf("%s",str1);

        printf("Enter Last Name : ");
        scanf("%s",str2);

        for(i=0;str1[i];i++); // iterate str1 till the last character

        str1[i++] = ' '; // insert space at next position
        for(j=0;str2[j];j++,i++)
        {
          str1[i] = str2[j]; //appending each character of str2 into str1
        }
        str1[i]='\0';
        printf("Hello %s",str1);
getch();
}
OUTPUT:
```



```
Enter First Name : Hiya
Enter Last Name : Jirte
Hello Hiya Jirte
```

5. *Convert str to uppercase.*

I. With function : strupr() easily does this task.

Syntax:

void strupr(char [])

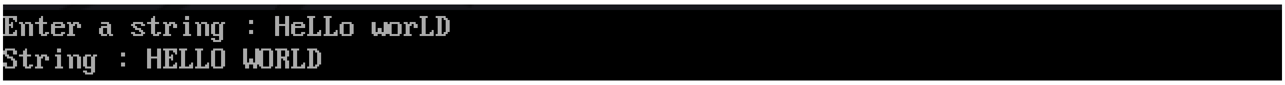
Example:

strupr(str)

Program 11: Write a program to convert String to uppercase.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char str[100];
clrscr();
        printf("Enter a string : ");
        gets(str);
        strupr(str);
        printf("String : %s\n",str);
getch();
}
```

OUTPUT:



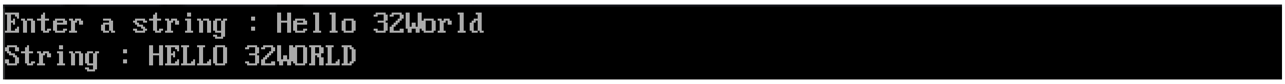
```
Enter a string : HeLLo worLD
String : HELLO WORLD
```

II. Without function : Difference between lower character & uppercase character is 32 (A:65 , a:97), so if a character is in lowercase then we need to subtract 32 from the character.

Program 12: Write a program to convert String to uppercase without function.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        char str[100];
        int i;
clrscr();
        printf("Enter a string : ");
        gets(str);
        for(i=0;str[i];i++)
        {
          if(str[i]>='a' && str[i]<='z')
               str[i] = str[i]-32;
        }
        printf("String : %s\n",str);
getch();
}
```

OUTPUT:



```
Enter a string : Hello 32World
String : HELLO 32WORLD
```

6. *Convert str to lowercase*

I. With function : strlwr() doest this task easily.

Syntax:

void strwlr(char [])

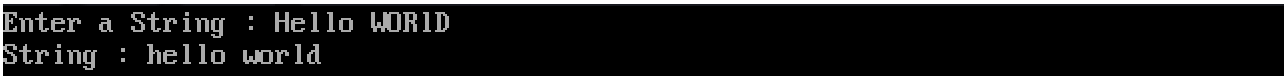
Example:

strlwr(str);

Program 13: Write a to convert string to lowercase with function.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char str[100];
clrscr();
        printf("Enter a String : ");
        gets(str);
        strlwr(str);
        printf("String : %s\n",str);

getch();
}
```

OUTPUT:



```
Enter a String : Hello WORlD
String : hello world
```

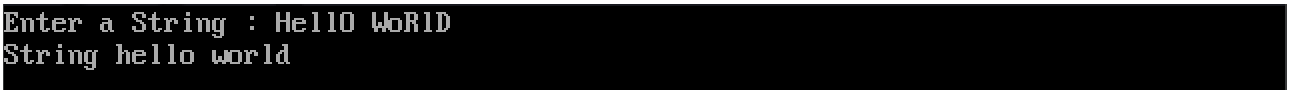
II. Without function : For uppercase to lowercase we need to add 32 into a character.

Program 14: Write a program to convert string into lowercase without function.

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

```
void main()
{     char str[100];
      int i;
clrscr();
       printf("Enter a String : ");
       gets(str);

       for(i=0;str[i];i++)
       {
          if(str[i]>='A' && str[i]<='Z')    //if character is in uppercase
          {
               str[i]=str[i]+32;
          }
       }
       printf("String %s\n",str);
getch();
}
```

OUTPUT:



```
Enter a String : HellO WoRlD
String hello world
```

7. *Finding a character in a String*

I. With function : strchr() helps us in this manner. Which return the address of that character if found it in the String. If the character is not there it will return 0.

Syntax:

char* strchr(char [],char)

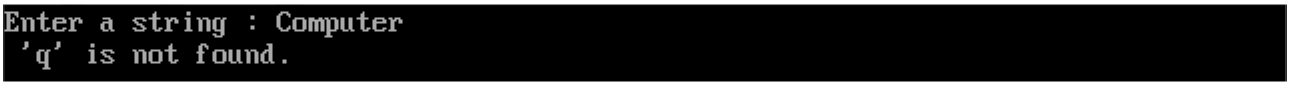
Example:

pos = strchr(str, 'A');

Program 15: Write a program to find the character 'q' in the given String.

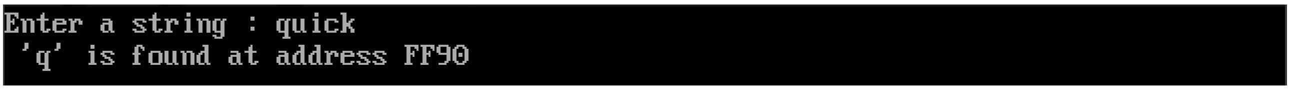
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
          char str[100],ch='q',*pos;
clrscr();
          printf("Enter a string : ");
          gets(str);
          pos = strchr(str,ch);
          if(pos)
            printf(" '%c' is found at address %p\n",ch,pos);
          else
            printf(" '%c' is not found.\n",ch);
getch();
}
```

OUTPUT 1:



```
Enter a string : Computer
 'q' is not found.
```

OUTPUT 2:



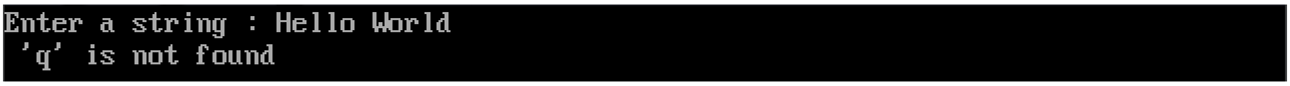
```
Enter a string : quick
 'q' is found at address FF90
```

II. Without function : first traverse(through looping) all character & check there is a character or not with if statement & if character is there store the address of that character in char pointer (char *) otherwise store 0 in that pointer.

Program 16: Write a program to find the character 'q' in the given string without function.

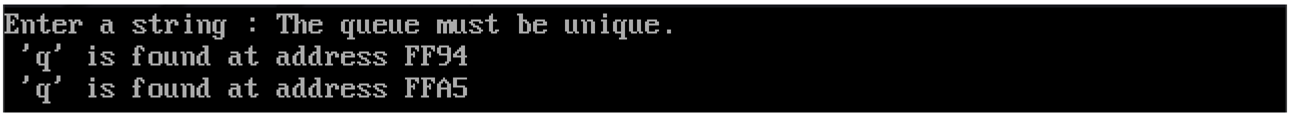
```
#include<stdio.h>
#include<conio.h>
void main()
{
        char str[100],ch='q',*pos=0;
        int i;
clrscr();
        printf("Enter a string : ");
        gets(str);

        for(i=0;str[i];i++)
        {
         if(str[i]==ch)
         {
           pos = &str[i];
           printf(" '%c' is found at address %p\n",ch,pos);
         }
        }

        if(!pos) //if pos points 0
         printf(" '%c' is not found\n",ch);

getch();
}
```

OUTPUT 1:



```
Enter a string : Hello World
 'q' is not found
```

OUTPUT 2:



```
Enter a string : The queue must be unique.
 'q' is found at address FF94
 'q' is found at address FFA5
```

NOTE : strchr() find only first occurred character not rest of them, but here we code to check every character.

8. Finding a substring in a String

I. With function : *strstr()* easily return the address of first character of sub-string in main string if found other wise returns 0.

Syntax:

char* strstr(char [],char [])

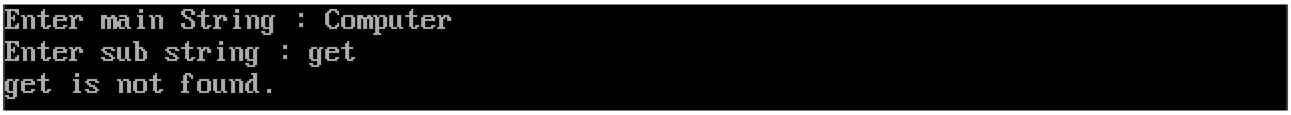
Example:

pos = strstr(str1,str2);

Program 17: Write a program to find sub-string in main string.

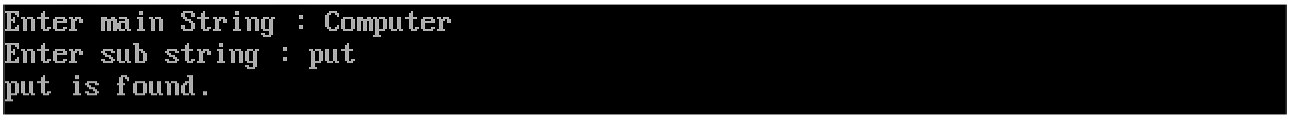
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main( )
{
        char str1[100],str2[100];
        char *pos;

clrscr( );
        printf("Enter main String : ");
        gets(str1);

        printf("Enter sub string : ");
        gets(str2);
        pos = strstr(str1,str2);
        if(pos)
         printf("%s is found.\n",str2);
        else
         printf("%s is not found.\n",str2);
getch( );
}
```

OUTPUT 1:



```
Enter main String : Computer
Enter sub string : get
get is not found.
```

OUTPUT 2:



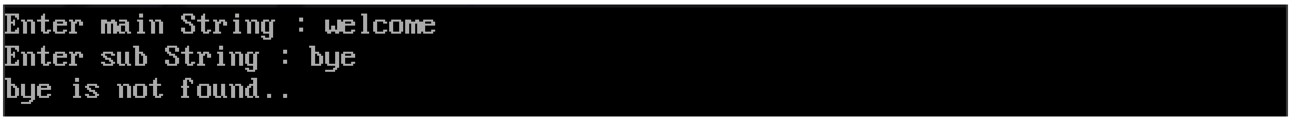
```
Enter main String : Computer
Enter sub string : put
put is found.
```

II. without function: take a flag variable with default value 0, find length of sub string then iterate all character of main string & check one bye one character of sub-string, if corresponding character is not matched reset index of sub string , at last during iteration if length of sub string is equals to value of index of sub-string that means we found the sub string in main string.

Program 18: Write a program to find sub-string in main string without using function.

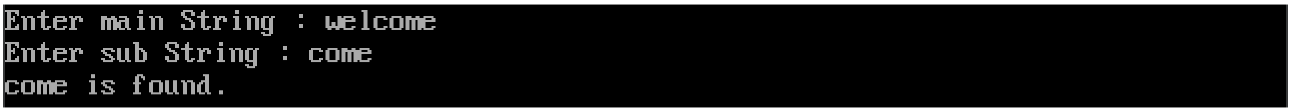
```
#include<stdio.h>
#include<conio.h>
void main()
{
        char str1[100],str2[100],*pos=0;
        int i,j,len,f=0;
clrscr();
        printf("Enter main String : ");
        gets(str1);

        printf("Enter sub String : ");
        gets(str2);
        for(len=0;str2[len];len++); // calculate length of str2
        for(i=0;str1[i];i++,j++)
        {
              if(j==len-1)
              {
                    f=1;
                   break; //exit from loop
              }
              if(str1[i]!=str2[j])
              {
                   j=-1;     // reset index of sub-string
              }
        }
        if(f) // if f==1
              printf("%s is found.\n",str2);
        else
              printf("%s is not found..\n",str2);
getch();
}
```

OUTPUT 1:



```
Enter main String : welcome
Enter sub String : bye
bye is not found..
```

OUTPUT 2:



```
Enter main String : welcome
Enter sub String : come
come is found.
```

9. *Reverse the String.*

I. With function : *strerev()* is used to reverse the String.

Syntax:

void strrev(char [])

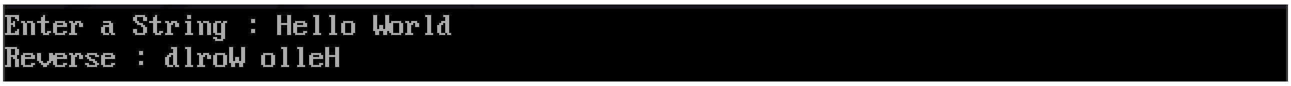
Example:

strrev(str);

Program 19: Write a program to reverse the String with help of strrev()

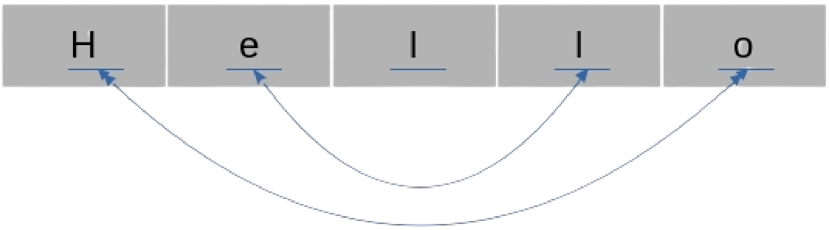
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main( )
{
        char str[100];
clrscr( );
        printf("Enter a String : ");
        gets(str);
        strrev(str); // reverse String
        printf("Reverse : %s\n",str);
getch( );
}
```

OUTPUT:



```
Enter a String : Hello World
Reverse : dlroW olleH
```

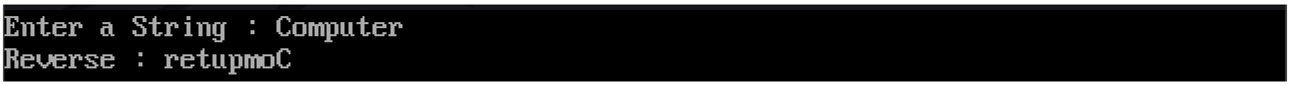
II. Without function : we can iterate all character from last & store those in another string makes reverse but it is time consuming process. So we will swap every starting character with ending with corresponding except middle one.



Program 20: Write a program to reverse the String without using strrev().

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        char str[100],temp;
        int i,j;
clrscr();
          printf("Enter a String : ");
          gets(str);

          for(j=0;str[j];j++); // length of str.
          j=j-1; // to avoid '\0'

          for(i=0;i<j;i++,j--)
          {
                temp = str[i];
              str[i] = str[j];
              str[j] = temp;
          }
          printf("Reverse : %s\n",str);
getch();
}
```

OUTPUT:



```
Enter a String : Computer
Reverse : retupmoC
```

10. Mask the String.

I. With function : Mask means rewrite all characters with a single character like password field.

Syntax:

void strset(char [],char)

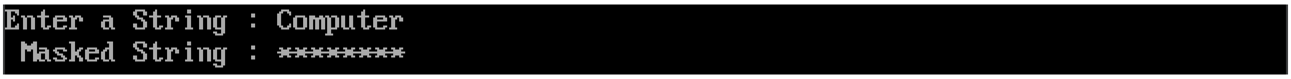
Example:

strset(str,ch);

Program 21: Write a program to mask the string with '@' sign.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
         char str[100],ch='*';
   clrscr();
         printf("Enter a String : ");
         gets(str);

         strset(str,ch);

         printf(" Masked String : %s\n",str);
   getch();
}
```

OUTPUT:



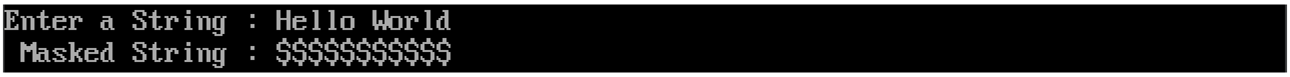
```
Enter a String : Computer
 Masked String : ********
```

II. Without function : This is very easy task to compare others. Just assign that character to all indexes.

Program 22: Write a program to mask the String with character '$';

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char str[100],ch='$';
   int i;
   clrscr();
        printf("Enter a String : ");
        gets(str);

        for(i=0;str[i];i++)
        {
          str[i] = ch;   // assign ch('$') to each character of str
        }
        printf(" Masked String : %s\n",str);
   getch();
}
```

OUTPUT:



```
Enter a String : Hello World
 Masked String : $$$$$$$$$$$
```

String array:

Technically String array is 2d array of char data type. Which is used to store multiple lines like a paragraph. To access specific line/string we need to provide index of that line. Which is start from 0 as normal array.

A	p	p	l	e	\0				
C	o	m	p	u	t	e	r	\0	
H	o	m	e	\0					
G	o	o	d		n	o	o	n	\0
W	o	r	k	\0					

as per table, there will be multiple lines all lines are terminate by \0 the NULL character.

Declaration :

Syntax:

char str_array[size1][size2];

example :

char words[10][100];

NOTE : here size1 determine how many lines be there & size2 determine how many characters should be in each line.

String array initialization

1. Compile time initialization :

   assign all values at declaration time.

Syntax:

char str_array[size1][size2] = {“str$_0$”, “str$_1$”, … “str$_{size1-1}$”};

Example:

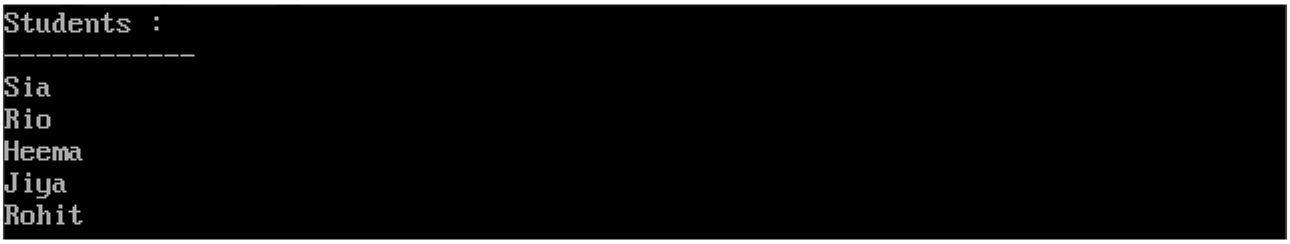
char words[4][100] = {“Hello”, “World”, “Parts”, “Peace”};

to access “World” we need syntax like words[1], here 1 is index of “World”

Program 23: Write a program to display name of 5 students.

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char names[5][100] = {"Sia", "Rio", "Heema", "Jiya", "Rohit"};
   int i,n=5;
   clrscr();
          printf("Students : \n");
          printf("------------\n");
          for(i=0;i<n;i++)
              printf("%s\n",names[i]);
   getch();
}
```

OUTPUT:



```
Students :
------------
Sia
Rio
Heema
Jiya
Rohit
```

2. Run time initialization:

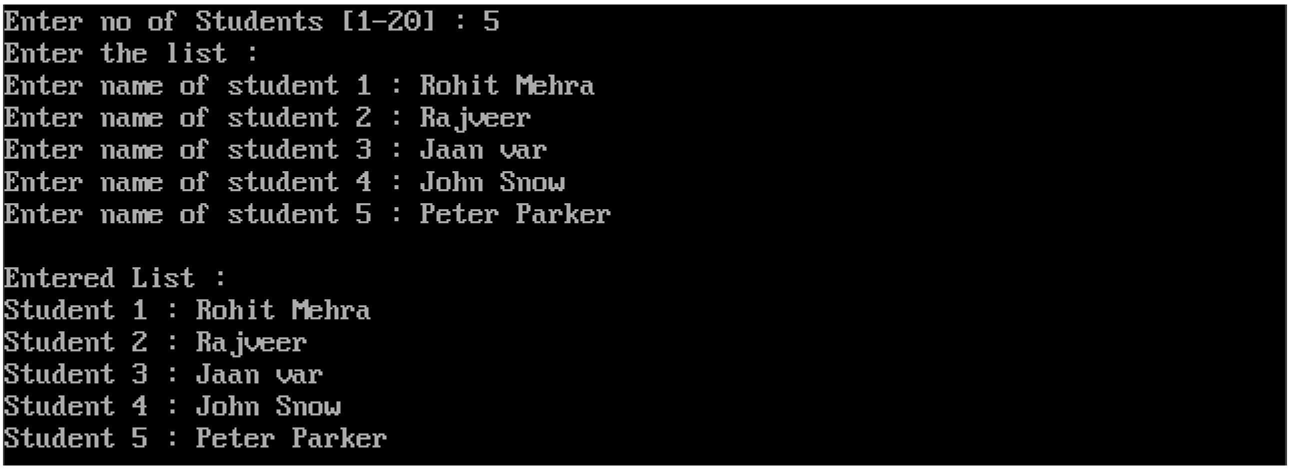
   To get data from user ,we can use scanf(), gets() in loop to get all data.

Program 24: Write a program to get a student list from user.

```
#include<stdio.h>
#include<conio.h>
void main()
{
   char list[20][100]; // max 20 students
   int i,n;
   clrscr();
     printf("Enter no of Students [1-20] : ");
     scanf("%d",&n);
       flushall(); // whenever we want to use gets after scanf,flushall needed
```

```
        printf("Enter the list : \n");
        for(i=0;i<n;i++)
        {
            printf("Enter name of student %d : ",i+1); //index+1
            gets(list[i]);
        }
        printf("\nEntered List : \n");
        for(i=0;i<n;i++)
            printf("Student %d : %s\n",i+1,list[i]);
    getch();
}
```

OUTPUT:

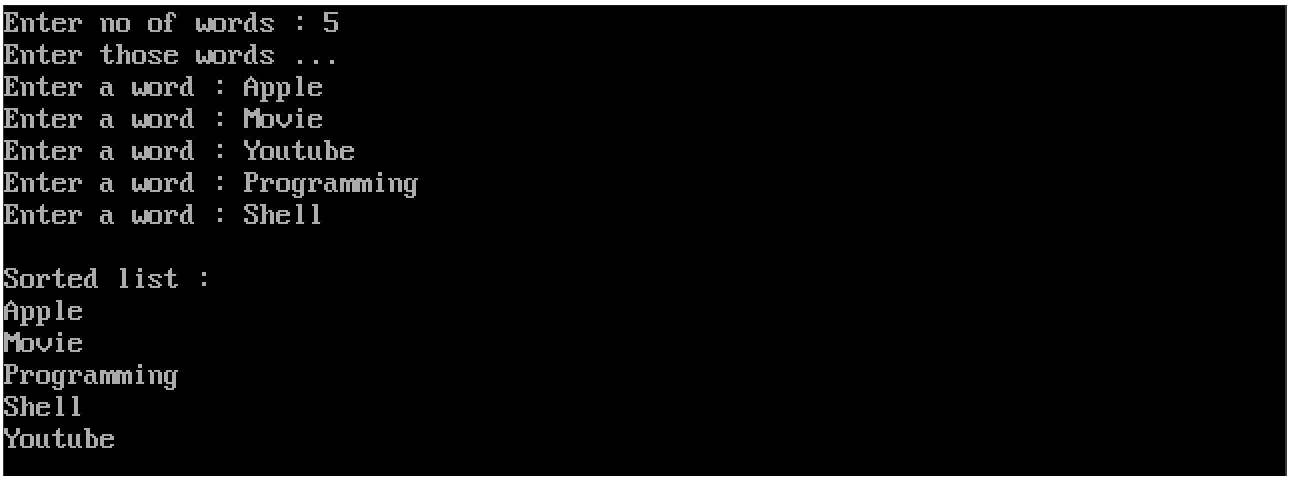


```
Enter no of Students [1-20] : 5
Enter the list :
Enter name of student 1 : Rohit Mehra
Enter name of student 2 : Rajveer
Enter name of student 3 : Jaan var
Enter name of student 4 : John Snow
Enter name of student 5 : Peter Parker

Entered List :
Student 1 : Rohit Mehra
Student 2 : Rajveer
Student 3 : Jaan var
Student 4 : John Snow
Student 5 : Peter Parker
```

Program 25: Write a program to sort words entered by user.

```
/*
        Here, we will compare all words with other ones, if strcmp()
returns +ve value then first word is higher than second one, so we can swap
it.
*/
#include<stdio.h>
#include<conio.h>
#include<string.h> // for strcmp(),strcpy()
void main()
{
    char words[20][100],temp[100];
    int i,j,n;
clrscr();
    printf("Enter no of words : ");
    scanf("%d",&n);
    printf("Enter those words ... \n");
        for(i=0;i<n;i++)
        {
          printf("Enter a word : ");
          scanf("%s",words[i]);
        }
        // sort logic
         for(i=0;i<n;i++)
         {
             for(j=i+1;j<n;j++) // i+1,target next one in array
             {
                if(strcmp(words[i],words[j]) > 0)
                {
                     strcpy(temp,words[i]);
                     strcpy(words[i],words[j]);
                     strcpy(words[j],temp);
                }
             }
         }
         printf("\nSorted list : \n");

         for(i=0;i<n;i++)
              printf("%s\n",words[i]);
getch();
}
```

OUTPUT:



```
Enter no of words : 5
Enter those words ...
Enter a word : Apple
Enter a word : Movie
Enter a word : Youtube
Enter a word : Programming
Enter a word : Shell

Sorted list :
Apple
Movie
Programming
Shell
Youtube
```