

Chapter 2 :- Strings

STRING :-

A String is a sequence of character terminated by a NULL character.

OR

String is a sequence of Characters.

TO COPY A STRING INTO ANOTHER STRING :-

What is we need to copy Content of One String to Another?

So we need to create another String but it should be blank Or Null String so it will be available to store characters.

So Suppose-

We have two strings

S1= Blank Or NULL String

S2= "Computer"

So we need to copy S2 into S1 letter by letter.

ALGORITHM:-

Step 1 : Start

Step 2 : Declare s1[10],s2[10],i

Step 3 : Input s1[10]

Step 4 :i=0

Step 5 : If s1[i]!=NULL then go to next step

Otherwise go to step 9

Step 6 : s2[i]=s1[i]

Step 7 :i++

Step 8 : Go to step 5

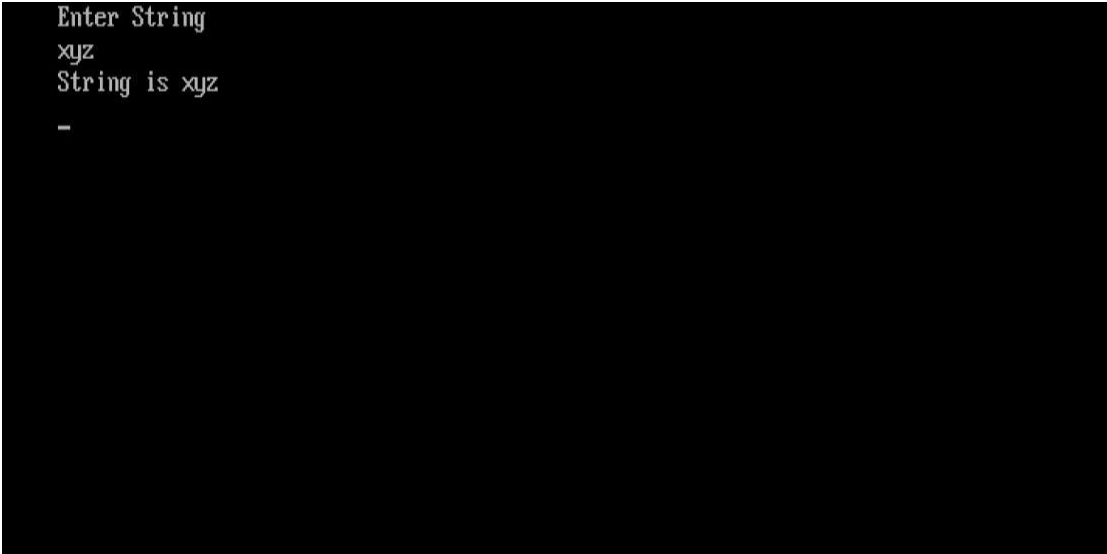
Step 9 : s2[i]=NULL

Step 10 : Print s2

Step 11 : END

PROGRAM :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    Int i;
    char s1[10],s2[10];
    clrscr();
    puts("Enter String");
    scanf("%s",s1);
        for(i=0;s1[i]!=NULL;i++)
        {
            s2[i]=s1[i];
        }
    s2[i]=NULL;
    printf("String is %s\n",s2);
    getch();
}
```

OUTPUTA screenshot of a terminal window with a black background and white text. The text shows the program's execution: 'Enter String' is printed, followed by the user input 'xyz' on the next line. Then 'String is xyz' is printed, and finally a single hyphen '-' is printed on the last line.

```
Enter String
xyz
String is xyz
-
```

TO CHECK WHETHER A GIVEN STRING IS PALINDROME OR NOT**What is a Palindrome String?**

As we all know string is a sequence of characters but if we reverse that string and if the string obtained after reversing the original string is same as the original string then that string is called Palindrome. For Ex- naman = naman (Palindrome) eye = eye (Palindrome) laptop != potpal (Not a Palindrome) book != koob (Not a Palindrome).

ALGORITHM:-

Step 1: Start

Step 2: Declare s[10] , s_rev[10] , i , Flag=0

Step 3: Input s[10]

Step 4: len=0

Step 5: if s[len]!=NULL then go to next step Otherwise go to step 8

Step 6: len++

Step 7: Go to Step 5

Step 8: i=0

Step 9: if s[i]!= NULL then go to next step Otherwise go to step 9

Step 10: s_rev[i]=s[len]

Step 11: i++, len - -

Step 12: Goto step 9

Step 13: s_rev[i]=NULL

Step 14: i=0

Step 15: if s[i]=s_rev[i] then go to next step Otherwise go to step 18

Step 16: Flag=1

Step 17: i++, Go to Step 15

Step 18: if Flag=1 then go to next step Otherwise go to Step 20

Step 19: Write(“String is Palindrome”) , Go to Step 21

Step 20: Write (“String is not Palindrome”)

Step 21: END

PROGRAM :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    Char s1[10],rev[10],rev1[10],i,j=0,flag=0,count=0,k;

    clrscr();

    //Input String from user//
    printf(“Enter String=”);
    gets(s1);
    for(i=0;s1[i]!='\0';i++)
    {

        count++; //Increment Count Until For Loop Reaches End of the String//

    }

    for(i=count-1;i>=0;i--)
    {

        rev[j]=s1[i]; //Reverse the String and Store it Character by Character in
        rev[j]// j++;
    }

    rev[j]='\0'; //Assign Null Value to rev[j]// printf(“Reversed
    String is %s\n”,rev); //Print the Reversed String//
    for(j=0;rev[j]!='\0';j++)
```

```
{  
rev1[k]=rev[j];    //Store Reversed String Temporarily in rev1[k]//  
}  
for(k=0;rev1[k]!='\0';k++)  
{  
if(rev1[k]==s1[k])    //Compare the Reversed String With Original String//  
{  
flag=1;    //Initialize flag to 1 if Reversed String and Original String are equal// }  
}  
if(flag==1)  
{  
printf("String is Palindrome");  
}  
else  
{  
printf("String is not Palindrome");  
}  
getch();  
}
```

```
Enter String=naman  
Reversed String is naman  
String is palindrome
```

TO COMPARE TWO STRINGS :-

In string comparison operation, if two strings S1 and S2 are given and if we compare S1 and S2 character by character, if both are same then given result "equal" and both are different then given result "not equal".

ALGORITHM :-

Step 1 : Start

Step 2 : Input s1[10],s2[10]

Step 3 : i=0

Step 4 : If s[i]!=NULL then go to next step

Otherwise go to step 7

Step 5 : i++

Step 6 : Go to step 4

Step 7 : x=i

Step 8 : i=0

Step 9 : If s[i]!=NULL then go to next step

```

        Otherwise go to step 7
Step 10 :i++
Step 11 : Go to step 9
Step 12 : y=i
Step 13 : If x!=y then
        Write("Strings are not Equal")
        Otherwise go to next step
Step 14 : i=0
Step 15 : If s1[i]!=s2[i] then
        Flag=1
        Otherwise i++ then Repeat step 15
Step 16 : if Flag=0 then
        Write("Strings are Equal")
        Otherwise go to next step
Step 17 : Write ("Strings are not Equal")
Step 18 : END

```


PROGRAM :-

```

#include<stdio.h>
#include<conio.h>
void main()
{
int i,a,b,l1,l2;
char s1[10],s2[10];
clrscr();
puts("Enter Two string");
scanf("%s %s",s1,s2);
for(i=0;s1[i]!=NULL;i++) ;
l1=i;
for(i=0;s2[i]!=NULL;i++);
l2=i;
if(l1!=l2)
{goto end1;

```

```
}  
for(i=0;s1[i]!=NULL;i++)  
{  
    if(s1[i]!=s2[i])  
    {  
        end1:  
        puts("String is not equal");  
        goto end;  
    }  
}  
puts("String is equal");  
end:  
getch();  
}
```

OUTPUT :-

```
Enter Two string  
xyz  
xyz  
String is equal  
-
```

TO CONVERT STRING FROM UPPER TO LOWER AND LOWER TO UPPER CASE :-

Converting a string from upper case to lower case by adding 32 in existing string and lower case to upper case by subtracting 32 from existing string.

ALGORITHM :-

Step 1: Start

Step 2: input s[10]

Step 3: initialize i=0

Step 4: if s[i]>=65 than go to next step otherwise go to step 7

Step 5: if s[i]<97 then go to next step otherwise go to step 7

Step 6: s[i]=s[i]+32

Step 7: if s[i]>=97 than go to next step otherwise go to step 10

Step 8: if s[i]<=122 than go to next step otherwise go to step 10

Step 9: s[i]=s[i]-32

Step 10: i++

Step 11: if s[i]==NULL than go to next step otherwise go to step 4

Step 12: print s

Step 13: end

PROGRAM :-

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

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

```
void main()
```

```
{
```

```
    inti;
```

```
    char s[10];
```

```
    clrscr();
```

```
    printf("Enter Uppercase string");
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i]!=NULL;i++)
```


```
    {
```

```
        if(s[i]>=65 && s[i]<97)
```

```
        {
```

```
        s[i]=s[i]+32;
    }
    else if(s[i]>=97 && s[i]<=122)
    {
        s[i]=s[i]-32;
    }
}

printf("Lower case string is\n%s",s);
getch();
}
```

OUTPUT :-

```
Enter Uppercase string
XYZ
Lower case string is
xyz_
```

STRING CONCATENATION :-

- Suppose we have two string S_1 and S_2 then concatenation of S_1 and S_2 are combined two strings together in one string. We required a concatenation function such as `xstrcat(S_1, S_2)`.

ALGORITHM:-

□ Procedure XSTRCAT(STR1,STR2,STR3)

Step 1: [Initialization]

STR3 NULL i

0

j 0 k

0

Step 2: [Copy STR1 string into STR3]

While(i<>LEN(STR1))repeat

(a) STR[k] STR[i]

(b) i i+1

(c) k k+1

(a) STR3[k] STR2[j]

(b) j j+1

(c) k k+1

Step 4: [Print the string after concatenation operation performed]

Write(“STR3”)

Step 5: [Finished]

Exit.

PROGRAM :-

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

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

```
void main()
```

```
{
```

```
Char str[10],str1[10],count=0;
```

```
Int i;
clrscr();
printf("Enter String 1:\n");
gets(str);
printf("Enter String 2:\n");
gets(str1);
    for(i=0;str[i];i++)
    {
        count++;
    }
for(i=0;str1[i];i++)
{
    str[count]=str1[i];
    count++;
}
str[count]=NULL;
printf("String is:\n");
puts(str);
getch();
}
```

OUTPUT

```
Enter String 1:
Shaival
Enter String 2:
Doshi
String is:
ShaivalDoshi
-
```

REVERSING A STRING**ALGORITHM :-**

REVERSE(string1,string2)

Step 1:- [Initialization]

i = 0

j = 0

Step2:-[To reach at end of original string]

While(i<>LEN(string1))

(i) i = i+1

Step3:-[store reverse string from original string]

(i) string2[j] = string1[i]

(ii) j = j+1

(iii) i = i-1

Step4:-[print the Reverse string]

Write(string2)

Step5:-[Finished]

Exit.

PROGRAM :-

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

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

```
Void main()
```

```
{
```

```
clrscr();
```

```
Void Reverse(char[],char[])
```

```
char str1[15],str2[15];
```

```
printf("\n enter original string:\n");
```

```
gets(str1);
```

```
str2='\0';
```

```
Reverse(str1,str2);
```

```
        Printf("Print Reverse string\n");
        Puts(str2);
        getch();
    }
void Reverse(char *s1,char *s2)
{
    int l;
    l=l-1;
    while(*s1 !='\0')
    {
    }
    while(l>=0)
    {
        *s2=*s1;
        s1--;
        s2++;
    }
    s2='\0';
}
```

OUTPUT

```
Enter String:
Smith
Reverse string is:
htims_
```