S12_1

String handling Functions

Objectives

To learn the following concepts

- String handling function

Session outcome

At the end of session student will be able to understand

- The String handling functions.
- Strcmp
- strcat

Library function: strcmp()

- The **strcmp function** compares two strings identified by the arguments and has a value 0 if they are equal.
- If they are not, it has the numeric difference between the first non matching characters in the strings.

strcmp(string1, string2);

string1 and string2 may be string variables or string constants.

e.g., **strcmp("their", "there");** will return a value of –9 which is the numeric difference between ASCII "i" and ASCII "r". That is, "i" minus "r" with respect to ASCII code is –9.

If the value is negative, string1 is alphabetically above string2.

Library function: strcat()

The **strcat function** joins two strings together.

It takes the following form:

strcat(string1, string2);

string1 and string2 are character arrays.

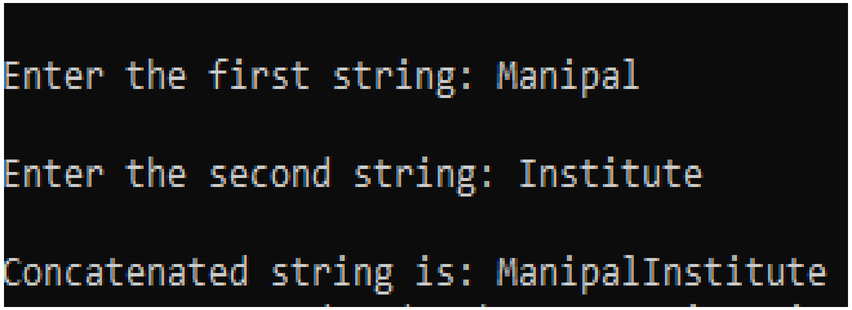
✓When the function **strcat** is excuted, string2 is appended to string1.

✓ It does so by removing the null character at the end of string1 and placing string2 from there.

✓The string at string2 remains unchanged.

Concatenation of 2 strings

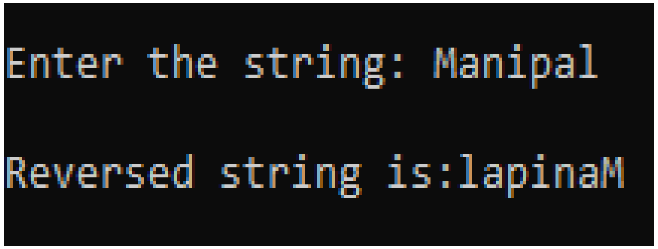
```
#include <stdio.h>

#include <string.h>

int main()

{     char s1[40], s2[50];

    printf("\nEnter the first string: ");

    gets(s1);

    printf("\nEnter the second string: ");

   gets(s2);

   strcat(s1, s2);

   printf("\nConcatenated string is: ");

    printf("%s",s1);

   return 0;   }
```



```
Enter the first string: Manipal

Enter the second string: Institute

Concatenated string is: ManipalInstitute
```

Reversing a string

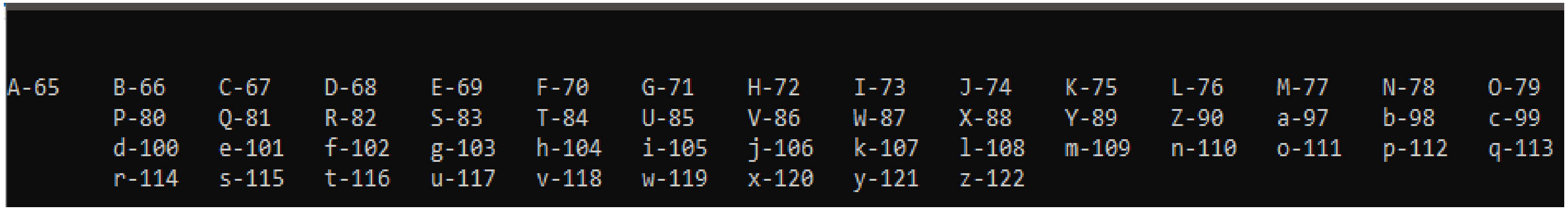
```
#include<stdio.h>
int main()
{
char str[70];
char temp;
int i, n=0;
printf("\nEnter the string:");
gets(str);
for(i=0;str[i]!='\0';i++)
   n++;
for(i=0;i<n/2;i++)
{
   temp=str[i];
   str[i]=str[n-i-1];
   str[n-i-1]=temp;
}
printf("\nReversed string is:");
puts(str);
return 0;
}
```



```
Enter the string: Manipal

Reversed string is:lapinaM
```

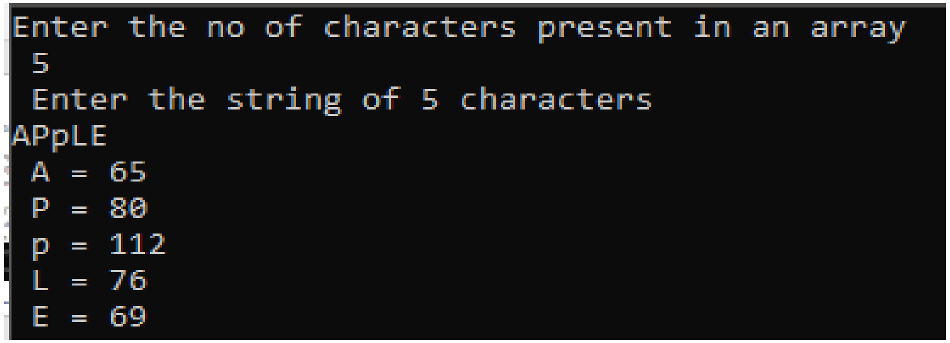
Print an alphabet in decimal [ASCII] & character form

```
#include<stdio.h>
int main()
{
char c;
printf("\n");
for(c=65;c<=122;c++)
{
 if(c>90 && c<97)
    continue;
 printf("%c", c);
printf("-");
printf("%d\t",(int)c);
}
printf("\n");
return 0;
}}}
```



```
A-65    B-66    C-67    D-68    E-69    F-70    G-71    H-72    I-73    J-74    K-75    L-76    M-77    N-78    O-79
        P-80    Q-81    R-82    S-83    T-84    U-85    V-86    W-87    X-88    Y-89    Z-90    a-97    b-98    c-99
        d-100   e-101   f-102   g-103   h-104   i-105   j-106   k-107   l-108   m-109   n-110   o-111   p-112   q-113
        r-114   s-115   t-116   u-117   v-118   w-119   x-120   y-121   z-122
```

Write a C Program to input a String & store their Ascii Values in an Integer Array & print the Array.

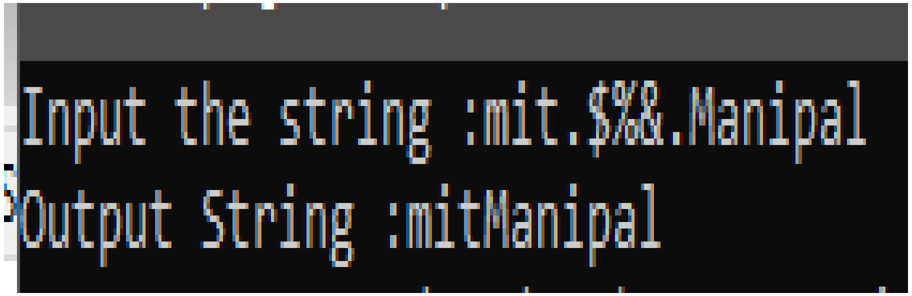


```
Enter the no of characters present in an array
 5
 Enter the string of 5 characters
APpLE
 A = 65
 P = 80
 p = 112
 L = 76
 E = 69
```

```
#include<stdio.h>
void main()
{ char string[20]; int asc[20];
int n, count = 0;
printf("Enter the no of characters present in an array \n ");
scanf("%d", &n);
printf(" Enter the string of %d characters \n" , n);
scanf("%s", string);
while (count < n)
{ asc[count]=string[count];
printf(" %c = %d\n", string[count], asc[count] );
++ count ;}}
```

Write a C program to remove special characters and digits leaving the alphabets un altered in a given string.

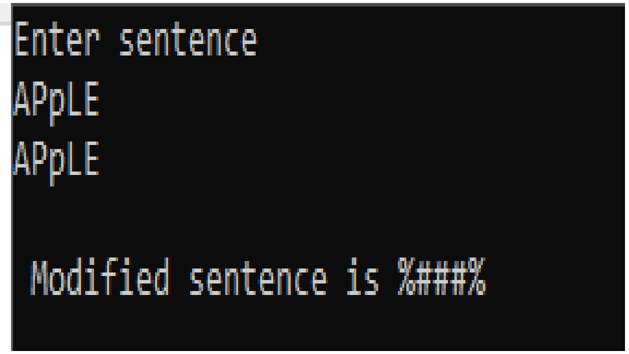
```
#include <stdio.h>
int main(){
char str[150];
int i,j;
printf("Input the string :");
scanf("%s",str);
for(i=0; str[i]!='\0'; ++i){
while (!((str[i]>='a'&&str[i]<='z') || (str[i]>='A'&&str[i]<='Z' ||
str[i]=='\0'))){
for(j=i;str[j]!='\0';++j){
str[j]=str[j+1];}
str[j]='\0'; }}
printf("Output String :%s", str);
return 0;}
```



```
Input the string :mit.$%&.Manipal
Output String :mitManipal
```

Write a C program to read a sentence and replace all the alphabets in the input sentence with ‘#’ whose ASCII value is even and with ‘%’, whose ASCII value is odd. Display the resultant sentence.

```
#include<stdio.h>
#include<string.h>
int  main()
{
const int Max = 100;
char sent[Max];
int i=0,count=0;
printf("Enter sentence \n");
gets(sent);
puts(sent);
while(sent[i]!='\0') {
  if( (sent[i]>='a'&& sent[i]<='z') ||
          sent[i]>='A'   && sent[i]<='Z')) {
   if(sent[i]%2==0)
     sent[i]='#';
   else
     sent[i]='%'; }
 i++; }
printf("\n Modified sentence is %s\n",sent);
return 0;}
```

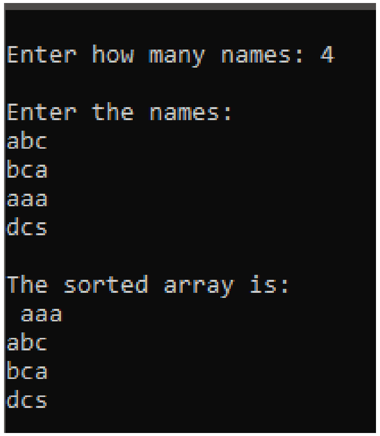


```
Enter sentence
APpLE
APpLE

 Modified sentence is %###%
```

Arrange ‘n’ names in alphabetical order

(hint: use string handling function-*strcpy*)

```
#include<stdio.h>
#include<string.h>
int main()
{
char a[10][10],temp[10];
int n,i,j;
printf("\nEnter how many names: ");
scanf("%d",&n);
printf("\nEnter the names: \n");
fflush(stdin);
for(i=0;i<n;i++)
gets(a[i]);
for(i=0;i<n-1;i++)
for(j=i+1;j<n;j++){
if(strcmp(a[i],a[j])>0){
strcpy(temp,a[i]);
strcpy(a[i],a[j]);
strcpy(a[j],temp);
}}
printf("\nThe sorted array is:\n ");
for(i=0;i<n;i++){
puts(a[i]);
}}
```



```
Enter how many names: 4

Enter the names:
abc
bca
aaa
dcs

The sorted array is:
 aaa
abc
bca
dcs
```

Tutorials on Simple Operations on String

- Write a simple C program to retrieve first word from a sentence.
- Write a C program to remove blank space from the string
- Write a C program to count the number of vowels and consonants in a given string.

Go to posts/chat box for the link to the question **PQn. S12.1**

submit your solution in next 2 minutes

The session will resume in 3 minutes

Summary

The String handling functions.

- strcmp

- strcat