INRODUCTION TO STRINGS IN PROGRAMMING LANGUAGE--C

PREPARED BY
SONIA MAHINDRU
ASSISTANT PROFESSOR IN COMPUTER SCIENCE
HANSRAJ MAHILA MAHA VIDYALAYA JALANDHAR

STRINGS

Strings are combination of characters

Syntax for displaying strings:

Printf(“\n HMV JALANDHAR\n”);

String is a character sequence stored as a one dimensional character array and terminated with a null character’\o’

Initializing string variables:

Initialization is done using character data type

char city[10];

This is used to read / write only one string having maximum length of 9 characters. Last character is null character.

char city[10][20];

This is used to read / write multiple strings. In this example 10 strings and every string has length not more than 19 characters.

Assigning data to string variable having constant value

Syntax:

Static datatype array name[size] = {list of values};

Static char city[10] = {‘B’,’O’,’M’,’B’,’A’,’Y’};

For multiple strings use two dimensional array

```
Static char city[3][20] = {     {‘B’,’O’,’M’,’B’,’A’,’Y’}
                                {‘D’,’E’,’L’,’H’,’I’}
                                {‘M’,’A’,’D’,’R’,’A’,’S’}     };
```

Reading and writing a string

For reading a single string

```
char city[10];
scanf("%s",city);
```

For reading multiple strings

```
char city[10][20];
for(i=0;i<10;i++)
{
scanf("%s",city[i]);
}
```

Reading and writing a string

For writing a single string

```
char city[10];
printf("%s",city);
```

For writing multiple strings

```
char city[10][20];
for(i=0;i<10;i++)
{
printf("%s",city[i]);
}
```

For reading two strings

```
char city1[10],city2[10];
scanf(“%s%s”,city1,city2);
```

For example to enter “NEW DELHI”

String Handling Functions

strcat( ): To concatenate or combine two different strings together.

```
strcat(string1,string2);
```

String1 is combined with string2 and result stored in string1.

Representation:

```
char string1[80],string2[30];
scanf("%s%s",string1,string2);
strcat(string1,string2);
printf("%s",string1);
```

Nesting:

```
strcat(strcat(string1,string2),string3);
```

strcmp( ): This function is used to compare two strings. It will check which string is alphabetically above the others. For comparison ASCII values are used.

```
strcmp(string1,string2);
```

If ASCII difference of each character in two strings from the first alphabet is **zero**, then both the strings are equal.

If difference is **+ve** then string2 is larger than string1.
If difference is **-ve** then string1 is larger than string2.

strcpy( ): This function is used to copy one string into another string. Target field should be larger than the source field. Size of string1 should be larger to receive the contents of string2.

strcpy(string1,string2);

Example:

```
char city[30];
strcpy(city,"BOMBAY");
strcpy(city1,city2);
```

Strlen(): This function is used to count the number of character in a string i.e to find the length of the string.

```
n=strlen(string);
```

n----------integer type

string variable----character type

Null character is not included in counting of characters.

Example:

```
int n;
char ct[20];
n=strlen(ct);
printf("length=%d",n);
```

Strrev():The purpose of this function is to reverse a string. This function takes string variable as its single argument.

strrev(st);

Program to find vowels in a given string

```
main()
{
int c ,i ,n;
Char a[20];
Clrscr();
Printf(“\nEnter the number of characters in the string:”);
Scanf(“%d”,&n);
Printf(“\nEnter the string:”);
For(i=1;i<=n;i++)
{
Scanf(“%c’”,&a[i]);
}
C=0;
For(i=1;i<=n;I++)
{
If(a[i]==‘a’ || a[i]==‘o’ || a[i]==‘u’ || a[i]==‘i’ || a[i]==‘e’)
{
c=c+1;
}
}
Printf(“\nNumber of vowels in astring is=%d”,c);
getch();
}
```

Program to check a string is palindrome or not

```
main()
{
   char text[20], reverse_text[20];
   int i,n, length = 0;

   printf("Enter text: ");
   gets(text);

   for (i = 0; text[i] != '\0'; i++)
   {
      length++;
   }
    for (i = length - 1; i >= 0; i--)
   {
      reverse_text[length - i - 1] = text[i];
   }
     for (n = 1, i = 0; i < length; i++)
   {
      if (reverse_text[i] != text[i])
      n = 0;
   }
```

```
 if (n == 1)
      printf("%s is a palindrome.", text);
   else
      printf("%s is not a palindrome", text);
getch();
   }
```

Program to reverse a string

```
void main()
{
    int i,n;
    char str[20];
    printf("Enter the String to get reversed\n");
    gets(str);
    n=strlen(str);
    printf("\nReversed string is \n");
    for(i=n-1;i>=0;i--)
    {
       printf("%c",str[i]);
    }

}
```

Program to concatenate two strings without using string handling function

```
void main()
{
  char str1[50],str2[50];
  static int i=0;
  int j=0;
  printf("\nEnter First String\n");
  gets(str1);
  printf("\nEnter Second String\n");
  gets(str2);
  while(str1[i]!='\0')
  {
    i++;
  }
   while(str2[j]!='\0')
  {
    str1[i]=str2[j];
    j++;
    i++;
  }
  str1[i]='\0';
  printf("\nConcatenated String is %s",str1);
}
```

Program to copy strings without using string handling function

```
main()
{
   char s1[100], s2[100];
   int  i;

   printf("\nEnter the string :");
   gets(s1);

   i = 0;
   while (s1[i] != '\0')
{
      s2[i] = s1[i];
      i++;
   }

   s2[i] = '\0';
   printf("\nCopied String is %s ", s2);

   getch();
}
```

Program to compare two strings

```
void main ()
{
  // declare variables
   char str1 [30], str2 [30];
   int i = 0, flag=0 ,len1, len2;

  printf ("Enter string1");
  gets (str1);
  printf ("\n Enter string2");
  gets (str2);
   len1 = strlen (str1);
   len2 = strlen (str2);
   while (i < len1 && i < len2 )
    {
      if( str1 [i] == str2 [i])
       {
         i++;
         continue;
       }
```

```
      if( str1 [i] < str2 [i])
      {
        flag = -1;
        break;
      }
     if( str1 [i] > str2 [i])
      {
        flag = 1;
        break;
      }
  }
  if (flag == 0)
    printf ("\n Both strings are equal ");
  if(flag == -1)
    printf ("\n string1 is less than string2 ");
  if( flag == 1)
    printf ("\n string1 is greater than string2 ");

getch();
}
```