C Programming

Class- BCA IInd Semester

Dr. Dharm Raj Singh

Assistant Professor, (HOD)

Department of Computer Application

Jagatpur P. G. College, Varanasi

Affiliated to Mahatma Gandhi Kashi Vidhyapith Varanasi

Email- dharmrajsingh67@yahoo.com

Outline

unit 3 : String

- Some of the String Functions in string.h
- strcpy() function
- strlen() function
- strcat() function
- strcmp() function
- To reverse a string without using strrev() function
- To convert string lower case to upper case without using string functions
- To c concatenate two Strings without using library function
- To search the occurrence of character in string

Strings

- The string is sequence of characters that is treated as a single data item. The character array ended with a null character \0 is implemented as string. A series of characters enclosed in double quotes (" ") is called a string constant, a string or string literal.
- The C compiler can automatically add a null character (\0) at the end of a string constant to indicate the end of the string. For example, "hello" is considered a string constant.

Declaring a string: C does not support string as the data type. However is allows us to represent strings as character arrays. It can be declared as – **char string_name[size];**

Initializing a String: strings can also be initialized at compile time and at run time. The process of initializing the strings at compile time is to write the string literal within double quotes. Compile time initialization can be done in three ways.

Type 1. char name[9] = “Sri Rama”;

Type 2. char stream[7] = {`H', `e', `l', `l', `o', `!', `\0'};

Type 3. char str[] = "I like C.";

Type 1: Consider a character array name that is initialized with a string constant “Mr JOHN”. This is written as

char name[9] = “Mr JOHN”;

This string is stored in the memory as given below.

M	r		J	O	H	N	\0	

- The compiler can automatically append a null character (\0) to the end of the array and treat the character array as a character string.
- Note that the size of the array is specified to hold up to nine elements, although the string constant has only eight characters enclosed in double quotes. The extra space is reserved for the null character that the compiler will add later.

- **Type 2:** array can be declared as string when null(\0) is appended at the end by user and array size is made sufficient to hold all characters. This is shown below.

  char stream[7] = {`H', `e', `l', `l', `o', `!', `\0'};/*string */

- **Type 3:** The third type of initialization is not specifying the array size. For example,

  char str[] = "I like C.";

```
#include <stdio.h>
int main ()
{
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
printf("Greeting message: %s\n",
greeting );
return 0;
}
```

Some of the String Functions in string.h

Function name	Description
strcpy(str1,str2)	copies str2 to str1 including the null (\0).
strcat(str1,str2)	Appends str2 to end of string str1
strlen(string)	Returns the length of string. Does not include the null(\0)
strcmp(str1,str2)	Compares str1 with str2 returns integer result.If str1<str2 returns negative integer.Returns zero when str1==str2,and Returns a positive integer when str1>str2.
Strncpy(str1,str2,n)	It copies at most n characters of str2 to str1.If str2 has fewer than n characters, it pads str1 with null('\o') characters.
Strchr(string,char)	Locates the position of the first occurrence of char within string and returns the address of the characterif it finds. and null if not.For ex-(" Hello", 'l')
strlwr()	converts all characters in a string from uppercase to lowercase.
strupr()	converts all characters in a string from lower case to uppercase.

Example of strcpy() function:

```
 int main ()
{
char str1[12] = "Hello";
char str2[12] = "World";
char str3[12];
/* copy str1 into str3 */
strcpy(str3, str1);
printf("strcpy( str3, str1) : %s\n", str3 );
return 0;
}
```

Example of strlen() function:

```
#include <stdio.h>
#include <string.h>
void main()
{
char name[30] = "This is string in C";
int len;
len= strlen( name );
printf( "Length of string <%s> is %d.\n", name ,len);
}
```

Example of strcat() function:

```
 int main ()
{
char str1[12] = "Hello";
char str2[12] = "World";
/* concatenates str1 and str2 */
strcat( str1, str2);
printf("strcat( str1, str2): %s\n", str1 );
return 0;
}
```

Example of strcmp() function:

```
main( )
{
char string1[ ] = "Jerry" ;
char string2[ ] = "Ferry" ;
int i, j, k ;
i = strcmp ( string1, "Jerry" ) ;
j = strcmp ( string1, string2 ) ;
k = strcmp ( string1, "Jerry boy" ) ;
printf ( "\n%d %d %d", i, j, k ) ;
}
```

Write a program to reverse a string without using strrev() function.

```
void main()
{
   char str[100], temp;
   int len, L,R;
   printf("Enter a string:");
   gets(str);
   len = strlen(str);
   L = 0;
   R = len -1;
   while(L < R)
   {
      temp = str[L];
      str[L] = str[R];
      str[R] = temp;
      L++;
      R--;
    }
     printf("Reversed string is: %s\n", str);
   getch();
}
```

Write a program to check given string is palindrome or not.

```
void main()
{
   char str1[25], str2[25];
   printf("Enter the string :");
   gets(str1);
   strcpy(str2,str1);
   strrev(str2);
    if (strcmp(str1,str2) == 0)
      printf("String is a palindrome ");
   else
      printf("String is not a palindrome ");
    getch();
}
```

Write a program check given string is palindrome or not without using string functions.

```
void main()
{
   char str[100];
   int start, middle, end, len = 0;
   printf("Enter the string :");
   gets(str);
    while (str[len] != '\0')
      len++;
    end = len - 1;
   middle = len/2;
   for (start = 0; start < middle; start++)
   {
      if (str[start] != str[end])
      {
         printf("String is not palindrome.\n");
         break;
      }
      end--;
   }
   if (start == middle)
      printf("String is palindrome.\n");

   getch();
}
```

Write a program to convert string lower case to upper case without using string functions.

```
#include <string.h>
 void main()
{
  char str[25];
  int i;
  printf("Enter any string:");
  scanf("%s",str);
  for(i=0;i<=strlen(str);i++){
          if(str[i]>=97&&str[i]<=122)
          str[i]=str[i]-32;
  }
  printf("\nThe string in lowercase is:",str);
  getch();
}
```

Write a program to c concatenate two Strings without using library function.

```
 void main()
{
 char str1[200], str2[200];
 int i, j;
 printf("Enter the first string: ");
 fgets(str1, 200, stdin);
 printf("Enter the second string: ");
 fgets(str2, 200, stdin);
 for(i=0;str1[i]!=NULL;i++)
 for(j=0;str2[j]!=NULL;j++)
 {
  str1[i++]=str2[j];
 }
 str1[i]=NULL;
 printf("Concatenated string is: %s",str1);
 getch();
}
```

Write a program to search the occurrence of character in string.

```
void main()
{
   char str[50], c;
   int count = 0, i;
   printf("\Enter a string : ");
   scanf("%s", &str);
   printf("\nEnter the character for searching : ");
   scanf("%c", &c);
   for (i = 0; str[i] != '\0'; i++) {
      if (str[i] == c)
         count++;
   }
    if (count == 0)
      printf("\n Character is not present", c);
   else
      printf("\n Occurrence of character '%c' is: %d", c, count);

   getch();
}
```

Exercise

1. Write a program to count the number of vowels in a given string.
2. Write a program to count the number of words in a given string. Two words are separated by one or more blank spaces.
3. Write a program that replaces two or more consecutive blanks in a string by a single blank.
4. Write a program to copy contents of one array into another array.
5. Read a string from keyboard and inverse the case of it. That is, convert lower case letters to upper-case and vice versa.
6. Write a program to find out the length of a given string without using the library function strlen().
7. Write a program to find out the length of a given string using the library function strlen().

References

- Kanetkar, Yashavant P. "Let Us C Fifth Edition." (2017).
- Kernighan, Brian W., and Dennis M. Ritchie. *The C programming language*. 2006.
- Ritchie, Dennis M., Brian W. Kernighan, and Michael E. Leask. The C programming language. Englewood Cliffs: Prentice Hall, 1988.
- McGraw-Hill, Herbert Schildt Tata. "The Complete Reference C fourth Edition”." (2005).
- Griffiths, David, and Dawn Griffiths. Head First C: A Brain-Friendly Guide. " O'Reilly Media, Inc.", 2012.
- Programming in C-Balguruswamy
- Structured programming approach using C-Forouzah & Ceilberg Thomson learning Publication

Declaration

The content is exclusively meant for academic purpose and for enhancing aching and learning. Any other use for economic/commercial purpose is strictly ohibited. The users of the content shall not distribute, disseminate or share it th anyone else and its use is restricted to advancement of individual owledge. The information provided in this e-content is authentic and best as r knowledge".

Dr. Dharm Raj Singh
Assistant Professor, (HOD)
Department of Computer Application
JagatpurP. G. College, Varanasi

Thanks