Module 13 – part 1 – String operations

BITS Pilani
Pilani Campus

Dr. Jagat Sesh Challa
Department of Computer Science & Information Systems

Module Overview

innovate achieve lead

- Strings
- Strings Operations

Strings

- Strings in C are represented by arrays of characters
- End of the string is marked with a special character NUL.
- The corresponding escape sequence character is \0.
- C does not have string data type.

Declaration of strings:

```
char str[30];
char line[80];
```

String Initialization


```
char str[9] = "I like C";
```

same as

```
char str[9]={'I',' ','l','i','k','e',' ','C','\0'};
```

Q. Is there any difference between following Initialization?

```
char str[]="BITS";
char str[4]= "BITS";
```

Ans: Yes, in second declaration there is no null character

Printing Strings

innovate achieve lead

```
char text[]="C Programming";
printf("%s\n",text);
```

Output???

```
C Programming
```

Reading a String (2-1)

innovate achieve lead

Using scanf()

```
char text[30];
printf("Enter a string: ");
scanf("%s",text);
printf("The string is : %s",text);
```

Sample output:

```
Enter a string: hello
The string is: hello
-------------------------------------
Enter a string: hello how are you
The string is: hello
```

Note: scanf() takes string without blank space

Reading a String (2-2)

```
char text[30];
printf("Enter a string: ");
scanf("%[a-z]s", text);
printf("The string is : %s",text);
```

Sample output:

```
Enter a string: hello
The string is: hello
-------------------------------------
Enter a string: hello123
The string is: hello
```

Single Line input

```
char text[80];
printf("Enter a string: ");
scanf("%[^\n]s",text);/*newline terminated string */
printf("The string is : %s",text);
```

Sample output:

```
Enter a string: hello how are you
The string is: hello how are you
```

Multi line Input


```
char text[180];
printf("Enter a string terminate with ~: ");
scanf("%[^~]s",text);
printf("The string is : %s",text);
```

Note: After ^ any character can be used to terminate the input.

Sample output:

```
Enter a string terminate with ~: hello how are
  you. ~
The string is: hello how are you.
```

Input String using gets()

innovate achieve lead

```
char str[200];
printf("Enter a string :");
gets(str);
printf("The string is :%s",str);
```

Output

```
Enter a string :C programming
The string is : C programming
```

Character Manipulation in the String – eliminate spaces

innovate achieve lead

```
int main()
{ char s[80],ws[80];
  int i,j;
  printf("Enter the text:\n");
  gets(s); /* reading text from user */
```

```
for(i=0,j=0; s[i]!='\0'; i++)
    {       if(s[i]!=' ')
            ws[j++] = s[i];
    }
ws[j]='\0';
```

```
  printf("The text without blank space is:\n");
  puts(ws); /* printing text on monitor */
  return;
}
```

Important Character functions in "ctype.h"

innovate achieve lead

```
isdigit(c)/*Returns a nonzero if c is a digit*/

islower(c) /* Returns a nonzero if c is a
              lower case alphabetic character */
isalpha(c)/*Returns a nonzero if c is an
             alphabet*/
isspace(c) /*Returns a nonzero for blanks */
isupper(c) /*Returns a nonzero if c is capital
             letter*/
toupper(c) /* Returns upper case of c */

tolower(c) /* Returns lower case of c */
```

String Manipulation functions in string.h

innovate achieve lead

```
strcpy(s1,s2) /* copies s2 into s1 */

strcat(s1,s2) /* concatenates s2 to s1 */

strlen(s) /* returns the length of s */

strcmp(s1,s2)/*returns 0 if s1 and s2 are same
             returns less then 0 if s1<s2
             returns greater than 0 if s1>s2 */
```

Other functions in string.h

innovate achieve lead

atof(): Converts an ASCII string to its floating-point equivalent (type double)	`char s1[] = "+1776.23";` `double my_value = atof(s1);`
atoi(): Converts an ASCII string to its integer equivalent	`char s2[] = "-23.5";` `int my_value = atoi(s2);`

strncat(): Works like strcat, but concatenates only a specified number of characters.	`char s1[50] = "Hello, world!";`
	`char s2[] = "Bye now!";`
	`strncat (s1, s2, 3);`

Other functions in string.h

innovate achieve lead

strncmp(): Works like strcmp, but compares only a specified number of characters of both strings.	`char s1[] = "dogberry";`
	`char s2[] = "dogwood";`
	`int comp = strncmp (s1, s2, 3);`

strncpy(): Works like strcpy, but copies only a specified number of characters.	`char dest[50];`
	`char src[] = "C Program";`
	`strncpy (dest, src, 3);`

strstr(): Tests whether a substring is present in a larger string. Returns a pointer to the first occurrence of the substring in the larger string, or zero if the substring is not present.	`char s1[] = "Got food?";`
	`char s2[] = "foo";`
	
	`if (strstr (s1, s2))`
	`printf("'%s' is a substring of '%s'.\n", s2, s1);`

Implementation of strlen()

```
int n = 0;
char text[100],c;
while((c = getchar())!='\n' && n<99)
      text[n++] = c;

text[n++]='\0';
printf("Length of text : %d",n);
```

Implementation of strcat()

innovate achieve lead

```
void main() /* s2 is concatenated after s1 */
{ char s1[100],s2[100];
  int i = 0,j = 0;
    printf("Enter first string\n");
    scanf("%[^\n]s",s1); getchar();
    printf("Enter second string\n");
    scanf("%[^\n]s",s2);
      while(s1[i++] != '\0'); i--;
      while(s2[j] != '\0')
        s1[i++] = s2[j++];
      s1[i] = '\0';
    printf("\n Final string is:%s",s1);
}
```

Palindrome problem

innovate achieve lead

```
void main()
{ char str[80];
  int left,right,i,len,flag = 1;
  printf("Enter a string");
  for(i = 0;(str[i] = getchar())!='\n';++i);
  len = i-1;
  for(left = 0,right = len; left < right; ++left,--right)
  {  if(str[left]!= str[right])
      { flag = 0;
        break;
      }
  }
  if(flag)printf("\n String is palindrome");
  else
  printf("\n String is not a palindrome");
}
```

Word counting Problem

innovate achieve lead

```
void main()
{ char text[40];
  int i = 0,count = 0;
  printf("Enter a string:");
  gets(text);
  while(text[i]!='\0')
     {
       while(isspace(text[i]))
       i++; /* Repeat till first non blank character */
       if(text[i]!='\0')
         {   count++;
             while(!isspace(text[i])&& text[i]!='\0')
             i++; /* Repeat till first blank character */

         }
     }
   printf("The number of words in the string is %d",count);
}
```

Arrays of Strings

innovate achieve lead

Declaration:

```
char name[5][30];
```

Five strings each contains maximum thirty characters.

Initialization:

```
char[5][10]={"One","Two","Three","Four","Five"};
```

Other valid declarations

```
char[][]={"One","Two","Three","Four","Five"};

char[5][]={"One","Two","Three","Four","Five"};
```

Array of Strings


```
char city[4][12] = {
  "Chennai",
  "Kolkata",
  "Mumbai",
  "New Delhi"
};
```

CLASSROOM

	0	1	2	3	4	5	6	7	8	9	10	11
0	C	h	e	n	n	a	i	\0				
1	K	o	l	k	a	t	a	\0				
2	M	u	m	b	a	i	\0					
3	N	e	w		D	e	l	h	i	\0		

dyclassroom.com

Array of Strings using Pointers


```
char *cityPtr[4] = {
  "Chennai",
  "Kolkata",
  "Mumbai",
  "New Delhi"
};
```


	0	1	2	3	4	5	6	7	8	9
0	C	h	e	n	n	a	i	\0		
1	K	o	l	k	a	t	a	\0		
2	M	u	m	b	a	i	\0			
3	N	e	w		D	e	l	h	i	\0

Exercise

```
int main () {

   char *cityPtr[] = {"Chennai", "Kolkata", "Mumbai",
  "New Delhi"};

   for (int i = 0; i < 4; i++)
      printf("\n cityPtr[%d] = %s", i, cityPtr[i] );

   return 0;
}
```

String Arrays: Reading and Displaying

innovate achieve lead

```
void main()
{ char name[5][30];
  printf("\n Enter five strings");
 /* Reading strings */
  for(i=0;i<5; i++)
  scanf("%s",s[i]);
 /* Printing strings */
  for(i=0;i<5; i++)
  printf("\n%s",name[i]);
}
```

Problem 3: Array of Strings

innovate achieve lead

Problem Statement: Write a C program that will read and store the details of a list of students in the format

ID NAME MARKS

And produce the following output

1. Alphabetical list of Names, ID's and Marks.
2. List sorted on ID's
3. List sorted on Marks

Implementation (1-3)

```
#define N 5
#include<stdio.h>
#include<string.h>
int main()
{
 char names[N][30],marks[N][10];
 char id[N][12],temp[30];
 int i,j;
/* Reading Student Details */
printf("Enter Student ID NAME and MARKS \n");
 for(i = 0; i<N; i++)
 scanf("%s %s %s",id[i],names[i],marks[i]);
```

Implementation (2-3)

```
/* Alphabetical Ordering of Names */
for(i=1; i<=N-1; i++)
    for(j=1; j<=N-i; j++)
      if(strcmp(names[j-1],names[j])>0)
        {  strcpy(temp,names[j-1]);
           strcpy(names[j-1], names[j]);
           strcpy(names[j], temp);
         /* Swapping of marks */
           strcpy(temp,marks[j-1]);
           strcpy(marks[j-1], marks[j]);
           strcpy(marks[j], temp);
```

Implementation (3-3)

```
  /* Swapping of ID's */
        strcpy(temp,id[j-1]);
        strcpy(id[j-1], id[j]);
        strcpy(id[j], temp);
      }
printf("ALPHABETICAL LIST OF ID NAME &
  MARKS");
for(i=0;i<N;i++)
printf("%s\t%s\t %s\n",id[i],names[i],marks[i]);
return;
}
```

ID wise Sorting (ID is a string)

```
for(i=1;i<=N-1;i++)
        {       for(j=1; j<=N-i; j++)
                 {  if(strcmp(id[j-1],id[j])>0)
                     { strcpy(temp,id[j-1]);
                       strcpy(id[j-1], id[j]);
                       strcpy(id[j], temp);
                  /* Swaping of marks */
                       strcpy(temp,marks[j-1]);
                       strcpy(marks[j-1], marks[j]);
                       strcpy(marks[j], temp);
                  /* Swaping of names */
                       strcpy(temp,names[j-1]);
                       strcpy(names[j-1], names[j]);
                       strcpy(names[j], temp);
                     }
                 }
        }
```

Home Exercise 1

innovate achieve lead

1) Write a C program to implement strcpy()

2) Write a C program to implement strcmp()

3) Write a C program to search a string from an array of strings.

4) Write a C program that copies the unique words among the set of words into another memory location.

5) Write a C program which will read a line of text and rewrite it in the alphabetical order.

6) Write a C program to replace a particular word by another word in a given string. Both the words are provided by the user at run time.

Home Exercise 2

1) Write a C program that counts the number of vowels, consonants, digits and other symbols in a given line of text.

2) Write a C program to reverse a string. Try not to use an extra string and modify the source string to store the reversed string. Number of exchanges should be minimal.

3) Write a C Program to separate a given string into two strings. All the odd positioned characters are stored in the first string and even positioned characters in the second string.

innovate achieve lead

BITS Pilani

Pilani Campus

Thank you

Q & A