

Strings

By P. Jayashree Reddy

Assoc. Prof.

CSE Dept.



JYOTHISHMATHI INSTITUTE OF TECHNOLOGY & SCIENCE

STRINGS

Definition:

A String is a variable length array of characters terminated by a null character('\0')

Declaration:

Syntax: `char variable-name[size];`

Eg: `char name[20];`

Maximum 19 characters can be stored & one null character

Initialization of strings

Like numeric arrays char arrays can be initialized

Eg1: `char str[20]="Jits Knr";`

Eg2: `char str[]="Jits Knr";`-----In this case the compiler will create an array of 9 bytes and initialize it with "Jits Knr" and a null character

We can also initialize a string as an array of characters eg:

`Char str[9]={'J','i','t','s','\0'};` Make sure that the null character is at the end.

Strings & the assignment operator

```
Char str1[6]="hello";
```

```
Char str2[6];
```

```
Str2=str1;----->compile error(an array name  
cannot be used as the left operand of an  
assignment operator
```

```
Char str[5];
```

```
Str="good";--->compile error(We cannot  
separate initialization from declaration)
```

String input/output statements

1) `scanf("%s",name);`----->It is used to accept a single word string meaning the i/p is terminated when blank space occurs in the i/p

Eg: `char name[10];`

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

If we give the i/p as jits knr the value of
`name=jits;`

2) gets(name)---->This function is used to accept multi word string meaning i/p is terminated when a newline char occurs

If the i/p is jits knr the value of name=jits knr

3)printf(“%s”,name);-----To display a string

The value of name is printed & the curser will wait in the same line for next o/p;

4)puts(name);-----After printing the o/p the curser goes to the next line;

CHATACTER FUNCTIONS

Char I/O functions

1) getchar();

Syntax:-ch=getchar();

It is used to read a single char from the key board
And press enter key to terminate.

2)ch=getche();

It is used to read a single char from the key board
But it does not need to an enter key to terminate
the i/p.

3) getch():--

It is used to read a single char but the given char is not displayed on the o/p screen

4) putchar()

Syntax:---putchar(ch);

It is used to display the given char on the o/p screen

Character handling functions

1) `islower(ch);` `returntype:---int`

It returns 1 if the given char is lower case letter
otherwise it returns 0

2) `isupper(ch);` `returntype:---int`

It returns 1 if the given char is upper case letter
otherwise it returns 0

3) `isdigit(ch);` `returntype:---int`

It returns 1 if the given char is digit otherwise it
returns 0

4) `isspace(ch);` `returntype:---int`

It returns 1 if the given char is space otherwise it returns 0

5) `toupper(ch);` `returntype:----char`

6) `tolower(ch);` `returntype:----char`

In order to use these character handling function, we have to include `ctype.h` header file.

Write a pgm to read a string of 1 line and display the number of uppercase letters,no of lowercase letters,no of digits, no of vowels,no of words and no of characters.At last display the given string into both uppercase and in lowercase.

```
#include<ctype.h>
main()
{
char line[81];
int i,wc,uc,lc,cc,vc,dc;
char ch;
printf("enter a string\n");
```

```
gets(line);
i=0;
wc=1;uc=0;lc=0;cc=0;vc=0;dc=0;
while(line[i]!='\0')
{
cc++;
ch=line[i];
if(isdigit (ch))
dc++;
If(isspace(ch))
wc++;
If(islower(ch))
lc++;
If(isupper(ch))
uc++;
If((tolower(ch)=='a')||(tolower(ch)=='e')||(tolower(ch)=='i')||(tolower(ch)=='
0')||(tolower(ch)=='u'))
vc++;
i++;
}
```

```
printf("Wordcount = %d",wc);
printf("Charcount = %d",cc);
printf("Digitcount = %d",dc);
printf("Uppercasecount = %d",uc);
printf("lowercasecount = %d",lc);
printf("vowelcount = %d",vc);
printf("Line in lowercase letters");
for(i=0;line[i]!='\0';i++)
printf("%c",tolower(line[i]));
printf("Line in uppercase letters");
for(i=0;line[i]!='\0';i++)
printf("%c",toupper(line[i]));
}
```

String Functions

1) `strcat()` : It joins two strings together.

Syntax : `strcat(string1,string2)`

Eg : `strcat(s1,s2)`

String s2 is appended to string s1.

String s2 remains unchanged.

2) strcmp() : It compares two strings .

Syntax : strcmp(string1,string2)

Eg : strcmp(s1,s2)

Comparison takes place char by char until a difference occurs or all characters are completed.If there is any difference it returns the difference b/w the ASCII values , otherwise it returns 0.

It returns 0 if both strings are equal

It returns -ve value if $s1 < s2$

It returns +ve value if $s1 > s2$

3) strcmpi() : It compares two strings .

Syntax : strcmpi(string1,string2)

Eg : strcmpi(s1,s2)

It is similar to strcmp() except both lowercase and uppercase letters are considered as same.

4) strlen() :

Syntax : strlen(string1)

Eg : strlen(s1)

It is used to find the length of the string excluding '\0'

5) strcpy() :

Syntax : strcpy(string1,string2)

Eg : strcpy(s1,s2)

It is used to copy one string into another string. String s2 is copied to string s1.

6) `strrev()` :

Syntax : `strrev(string1)`

Eg : `strrev(s1)`

It is used to reverse the given string.

7) `strncpy()` : Length controlled string copy

Syntax : `strncpy(string1,string2,Size)`

Eg : `strcpy(s1,s2,5)`

It copies first 5 characters of the string s2 into string s1.

Note: Since the first 5 characters may not include the terminating null character, we have to place it explicitly in the 6th position of s2.

`s1[5]='\0'`

8) `strncmp()` :

Syntax : `strncmp(string1,string2,n)`

Eg : `strncmp(s1,s2,n)`

It compares the left most n characters of s1 to s2.

9) `strncat()` :

Syntax : `strncpy(string1,string2,Size)`

Eg : `strncpy(s1,s2,5)`

This will concatenate the left most n characters of s2 to the end of s1

8) `strncmp()` :

Syntax : `strncmp(string1,string2,n)`

Eg : `strncmp(s1,s2,n)`

It compares the left most n characters of s1 to s2.

9) `strncat()` :

Syntax : `strncpy(string1,string2,Size)`

Eg : `strncpy(s1,s2,5)`

This will concatenate the left most n characters of s2 to the end of s1

10) strstr() :

Syntax : strstr(string1,string2)

strstr() searches the string s1 to see whether string s2 is contained in s1.If yes the function returns the position of the first occurrence of the substring.Otherwise it returns a Null

Eg: if(strstr(s1,s2)==null)
 printf("substring not found");
 else
 printf("s2 is a substring of s1");

11) strchr() : Determines the existence of a character in a string.

Syntax : strchr(string1,character to be located)

Eg : strchr(s1,'m')

It will locate the first occurrence of the character 'm' in the string s1

12) strrchr() : Determines the existence of a character in a string from right.

Syntax : strrchr(string1,character to be located)

Eg : strrchr(s1,'m')

It will locate the last occurrence of the character 'm' in the string s1

13) strspn() : String span

Syntax : strspn(string1,string2)

Eg : s1="concatenation"

len=strspn(s1,"aeioucn");

The value of len =5

It searches the string spanning characters that are in the set and stopping at the first character that is not in the set.It returns the number of characters that matched those in the set.If no characters match those in the set , then they return 0.

14) strcspn() : Complemented String span

Syntax : strcspn(string1,string2)

Eg : s1="concatenation"

len=strcspn(s1,"teibx");

The value of len =5

It functions stop at the first character that matches one of the characters in the set.

15) strtok() : String token

Syntax : strtok(string1,delimiter)

Eg : strtok(s1," ")

The string s1 will be split into tokens when it comes across a space.

16) strtol() : String to long

Syntax : strtol(string1,string2,int base)

s2---> is pointer which holds the
address of the trailing string

s1---> string to be converted to long

base---> may be 0 or 235.

Eg : num=strtol("12345DC:",&ptr,10);

num=12345

ptr---> "DC"

It converts a string to a long integer. It skips leading whitespace characters and stops with the first non-numeric character, which is considered to be the start of a trailing string.

17) strtod(): String to double

Syntax : strtod(string1,*string2)

Eg : strtod(s1,*s2)

It is similar to strtol() except that it does not have a base

18) Other number functions:

strtof() -> string to float

strtold() -> string to long double

strtoll() -> string to long long int

strtoul()-> string to unsigned long int

Write a program to print the given string in the following format(cprogramming)

c
cp
cpr
cpro
.
.
cprogramming
cprogrammin
cprogrammi
.
.
cp
c

```
main()
{
char lang[13]="cprogramming";
int l;
for(i=1;i<=12;i++)
{
printf("\n%-12.*s",i,lang);
}
for(i=12;i>=1;i--)
{
printf("\n%-12.*s",i,lang);
}
}
```

Program to check given string is palindrome or not

```
#include<string.h>
main()
{
    char p[30];
    int i,d,l,mid;
    printf("enter a string");
    scanf("%s",p);
    l=strlen(p);
```

```
mid=l/2;
for(i=0,d=l-1;i<=mid;i++,d--)
{
if(p[i]==p[d])
    continue;
else
{
printf("not palindrome");
exit();
}
}
printf("palindrome");
}
```

Program to reverse a string without using library functions

```
main()
{
    char p[30],a[30];
    int c=0,i=0,j=0;
    printf("enter a string");
    scanf("%s",p);
    while(p[i]!='\0')
    {
        c++;
        i++;
    }
```

```
for(i=c-1;i>=0;i--)  
{  
    a[j]=p[i];  
    j++;  
}  
a[j]='\0';  
printf("given string=%s",p);  
printf("reverse string=%s",a);  
}
```

Program to sort a given string

```
main()
{
char s[30],c;
int i,j,k;
printf("enter a string");
gets(s);
k=strlen(s);
for(i=0;i<k-1;i++)
{
for(j=i+1;j<k;j++)
{
```

```
if(s[i]>s[j])
{
    c=s[i];
    s[i]=s[j];
    s[j]=c;
}
}
}
printf("sorted string");
puts(s);
}
```

Array of Strings

- To store multiple strings we use two dimensional character arrays.
- They are called as array of strings.
- First dimension indicates the no. of strings and the second dimension indicates the maximum length of each string.
- Eg. `char names[10][15];`
- We can store 10 names where the maximum size of each name is 14 characters.

- We can access the values from these strings character by character by using %c with two dimensions or we can access string by string by using %s and specifying only one dimension .

Program to sort the names in alphabetical order.
The list of names should be asked

```
main()
{
char names[10][10],ch,t[10];
int i,j,k=0;
do
{
printf("enter a name");
scanf("%s",names[k]);
fflush(stdin);
ch=getchar();
k++;
}while((ch=='y')||(ch=='Y'));
```

```
for(i=0;i<k-1;i++)
{
for(j=i+1;j<k;j++)
{
if(strcmp(names[i],names[j])>0)
{
strcpy(t,names[i]);
strcpy(names[i],names[j]);
strcpy(names[j],t);
}
}
}
```

```
for(i=0;i<k;i++)  
    printf("%s",names[i]);  
}
```