

STRING



- String is collection of Characters (or) group of elements in which symbols enclosed within quotation marks.
- String is always declared as character arrays.
- in other words character arrays are called as STRINGS.

```
char name[5]={'l','n','d','l','a'};
```

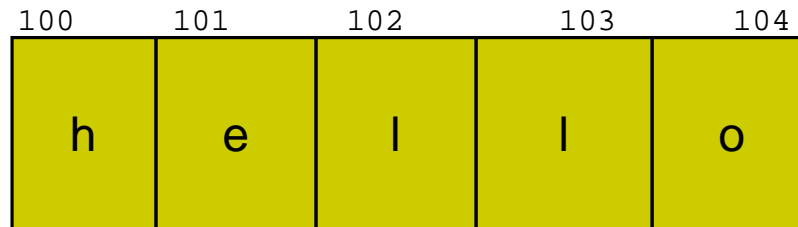


- interpretation

Arrays whose elements are characters called as string

- Strings are always terminated with a NULL character ('\0' or 0)

```
char a[]="hello\n"; /*size?*/
```



String Initialization



- Initialization
 - `char m[9] = "I like C";`
 - `char m[] = "I like C";`
 - `char m[] = { 'I', ' ', 'l', 'i', 'k', 'e', ' ', 'C' };`
 - `char m[] = { { 'I' }, { 'l' }, { 'i' }, { 'k' }, { 'e' }, { 'u' } };`

```
main()
{
char name1[9] = "I like C";
char name2[9 ] = { { 'l' },{ 'l' },{ 'i' },{ 'k' },{ 'e' },{ 'u' } };
char name3[9 ] = { 'l', ' ', 'l', 'i', 'k', 'e', ' ', 'c', '\0' };
clrscr();
printf("name1=%s",name1);
printf("name1=%s",name2);
printf("name1=%s",name3);
}
```



Print the elements of char array



```
main()
{
char str[15]="have a nice day";
int i=0;
while(i<=15)
{
printf("%c",str[i]);
i++;
}
}
```

Print the elements of char array



```
main()
{
char str[ ]="have a nice day";
int i=0;
while( str[i]!='\0')
{
printf("%c",str[i]);
i++;
}
}
```

Standard string functions



- Strlen :-determines length of string
- Strcpy :-copies a string from source to destination
- Strncpy:-copies char of string to another string upto specific length
- Strcmp:-compare char of 2 strings
- Stricmp:-compare 2 strings(NOT case sensitive).
- Strncmp:-compare char of 2 strings upto specific length
- Strnicmp:-compare char of 2 strings upto specific length .Ignore case.

Strlen function



It counts the number of characters in a given string.

To count no of chars in a given string



```
main()
{
char str[10];
int length;
printf("enter string");
gets(str);
length=strlen(str);
printf("length of string=%d",length);
}
```

Strcpy function



This function copies the contents of 1 string to another.

```
strcpy(s2,s1);
```

S1 =source string

S2 =destination string

S1 is copied to s2.

To copy contents of 1 string to other



```
main()
{
char s1[10],s2[10];
printf("enter string");
gets(s1);
strcpy(s2,s1);
printf("first string",s1);
printf("second
string",s2);
}
```

```
main()
{
char s1[10],s2[10];
int i;
printf("enter string");
gets(s1);
for(i=0;i<10;i++)
s2[i]=s1[i];
printf("first string",s1);
printf("second string",s2);
}
```

Copy contents upto a specific length



```
main()  
{  
    str1[10],str2[10];  
    int n;  
    printf("enter source string");  
    gets(str1);  
    printf("enter destination string");  
    gets(str2);  
    Printf("enter no. of char to be replaced");  
    Scanf("%d",&n);  
    strncpy(str2,str1,n);  
    printf("first string",str1);  
    printf("second string",str2);  
}
```

///strncpy fun//

Stricmp function



- This function compares 2 strings. it compares 2 strings **without knowing** upper case and lower case. if strings are same then it returns to 0 otherwise non-zero value.

```
diff =stricmp(str1,str2);  
if(diff == 0)  
puts("strings are equal");  
else  
puts("strings are not equal");
```



Strcmp function

- This function compares 2 strings. it compares 2 strings and also check the upper case and lower case. if strings are same then it returns to 0 otherwise non-zero value

```
diff =strcmp(str1,str2);  
if(diff= =0)  
puts("strings are equal");  
else  
puts("strings are not equal");
```

Strncmp function

strncmp(source,target,argument)

```
main()
{
char sor[10],tar[10];
int n,diff;
printf("enter first string");
gets (sor);
printf("enter second string");
gets(tar);
printf("enter length upto which comp is made");
scanf("%d",&n);
diff =strncmp(sor,tar,n);
if (diff ==0)
puts("strings are same upto %d characters",n);
else
puts("two strings are different");
}
```



Strlwr and strupr



```
main()
{
char a[15];
printf("enter string in upper
      case");
gets(a);
printf("in lower case string is:-
      %s",strlwr(a));
}
```

```
main()
{
char a[15];
printf("enter string in lower
      case");
gets(a);
printf("in upper case string
      is:-%s",strupr(a));
}
```


Strdup function



It is used for duplicating a string at the allocated memory which is pointed by pointer variable.

```
main()
{
char text1[10], *text2;
printf("enter text");
gets(text1);
text2=strdup(text1);
printf("original string :%s \n duplicate strin:%s",text1,text2);
}
```

Strchr function

It returns the pointer to a position in the first occurrence of the character in given string.



```
main()
{
char str[20],ch,*p;
printf("enter text");
gets(str);
printf("enter text to find");
ch=getchar();
p=strchr(str,ch);
if(p)
printf("char %c found in string",ch);
else
printf("char %c not found in string",ch);
```



Strcat, strncat function

```
main()
{
char s1[10],s2[10];
puts("enter text 1");
gets(s1);
puts("enter text 2");
gets(s2);
strcat(s1,s2);
printf("%s",s1);
}
```

```
main()
{
char s1[10],s2[10],n;
puts("enter text 1");
gets(s1);
puts("enter text 2");
gets(s2);
puts("no of char to add");
gets(n);
strcat(s1," ");
strncat(s1,s2,n);
printf("%s",s1);
}
```



More functions

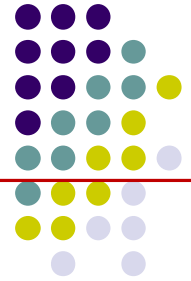
- `strrev()`:-it simply reverse the string.
`puts(strrev(str));`
- `Strset()`:- it replaces every char of string with symbol given by programmer.
- `Strnset()`:-same as `strset` function but in replaces only 'n' no. of chars.
- `Strspn()`:-it returns the position of the string where source array doesn't match with target one.
- `Strpbrk()`:-it searches the first occurrence of the char in a given string and then display the string from that character.

Atoi() function



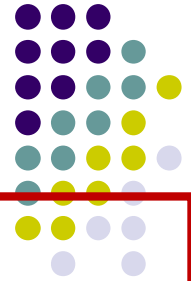
```
main()
{
int z,t;
char a[20];
x="1998";
z=atoi(x);
printf("%d",z);
t=z-1;
printf("%d",t);
}
```

Count Length Of String



```
#include<stdio.h>
#include<conio.h>
void main()
{
    char ch[10];
    int i=0;
    gets(ch);
    While(ch[i]!='\0')
    {
        i++;
    }
    printf("%d",i);
    getch(); }
```

Copy a String



```
#include<stdio.h>
#include<conio.h>
void main()
{
    char ch[10],bh[10];
    int i=0;
    gets(ch);
    while(ch[i]!='\0')
    {
        bh[i]=ch[i];
        i++;
    }
    bh[i]='\0';
    puts(bh);
    getch();
}
```

Sort List of elements in desc



```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10],l,j,n,temp;
    printf("\nEnter the no of elements\n");
    scanf("%d",&n);
    for(i=0;i<=n-1;i++)
        scanf("%d",&a[i]);
    for(i=0;i<=n-1;i++)
    {
        for(j=0;j<=n-i-1;j++)
        {
            if(a[j+1]>a[j])
            {
                temp=a[j+1];
                a[j+1]=a[j];
                a[j]=temp;
            }
        }
    }
}
```




Array of pointer to string

```
char *t[20] , a[20][30];  
int i;  
for(i=0;i<30;i++)  
{  
t[i]=a[i];  
}
```

As the variable 'a' is itself an array of type char,so an array of pointers to strings 't' is an array of pointer to char.

Array of pointer to string



```
main()
{
char *city[ ]={"ludhiana","cantt","jalandhar"};
int l;
for(i=0;i<3;i++)
{
printf("%s city has address %u",*(city+i),(city+i));
}
```

Pass Array of strings to function



- Array of strings can be passed to a function as arguments. formal arguments can be declared as a pointer to a type character. it can be representing as:-

```
char **v;
```

Pass Array of strings to function



```
main()
{
char *city[ ][ ]={"ludhiana","cantt","jalandhar"};
void pass(char**,int);
pass(city,3);
getch();
}
void pass(char **m,int n)
{
int i;
for(i=0;i<n;i++)
{
printf("%d".m[i]);
}
}
```



- We can not assign a string to another string, whereas we can assign a **char pointer to another char pointer**.

```
main()
{
char str1[ ]="hello";
char str2[15];
char *s="how are you";
char *p;
str2=str1;          /*error*/
p=s;               /*works*/
getch();
}
```