Strings
(character array)

Strings

Just like ***int*** and ***float***,

char is a data type which is used to store the value of a character.

```
int main()
{
    char ch;
    scanf("%c",&ch);
    printf("%c",ch);
}
```

```
int main()
{
    char ch='h';
    printf("%c",ch);//h
}
```

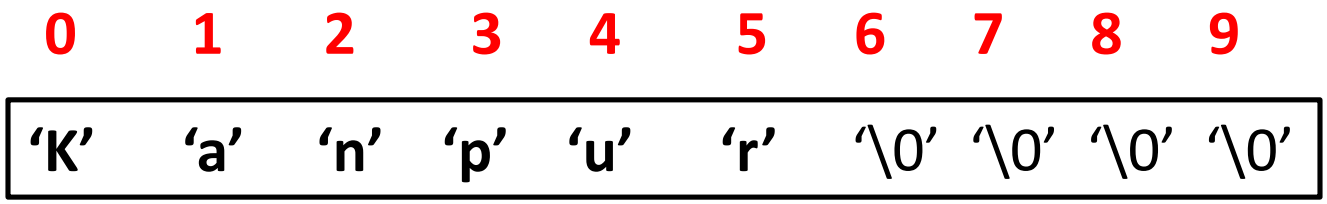
A string (word /sentence) is a collection of characters.
In 'C' ***char array*** is used to represent a string.

Strings

- String is an array of characters.
- Declaration of string(char array):

  char s[10];
- declaring a string(word) of 10 characters.
- Initializing a string:

  char name[10]="Kanpur";



Every string is terminated with NULL chracter('\0').

Strings

- String is an array of characters.
- Declaration of string(char array):

  char s[20]; **// declaring a string(word) of 20 characters**

- Initializing a string:

  char name[]="PSIT Kanpur"; **// char name[]="PSIT Kanpur";**

  or

  char name[12]={'P', 'S', 'I', ' T', ' ', 'K', 'a' 'n', 'p', 'u', 'r', '\0' };

0	1	2	3	4	5	6	7	8	9	10	11
'P'	'S'	'I'	'T '	' '	'K'	'a'	'n'	'p'	'u'	'r'	'\0'

Every string is terminated with NULL chracter('\0').

Strings

- String is an array of characters.
- Declaration of string(char array):

  char s[20]; **// declaring a str**

- Initializing a string:

  char name[]="PSIT Kanpur"; **//**

  or

  char name[12]={'P', 'S', 'I', ' T', ' ', 'K', 'a' 'n', 'p', 'u', 'r', '\0' };

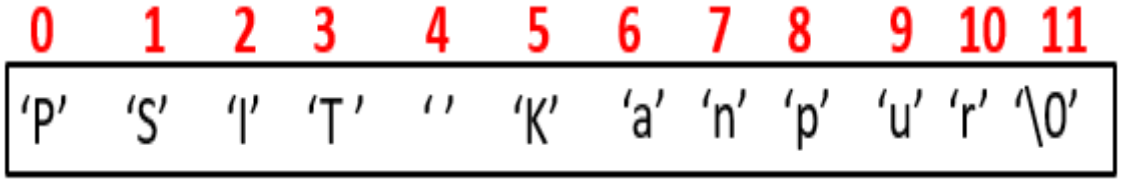
Important: there is an extra terminating character which is the Null character ('\0') used to indicate the termination of string which differs strings from normal character arrays.

0	1	2	3	4	5	6	7	8	9	10	11
'P'	'S'	'I'	'T '	' '	'K'	'a'	'n'	'p'	'u'	'r'	'\0'

Every string is terminated with NULL chracter('\0').

Strings

- **Initializing a string:**



char name[12]="PSIT Kanpur";

or

char name[]="PSIT Kanpur";

or

char name[12]={'P', 'S', 'I', ' T', ' ', 'K', 'a' 'n', 'p', 'u', 'r', '\0' };

or

char name[]={'P', 'S', 'I', ' T', ' ', 'K', 'a' 'n', 'p', 'u', 'r', '\0' };

Strings

Taking string from user:

char str[20];

scanf("%s",str);

As str[] is a character array so using str without braces '[' and ']' will give the base address of this string. So we need not use '&' with scanf().

```
main()
{
   char str[20];
   printf("enter ur string: ");
   scanf("%s", str); // do not use   '&str' in scanf()

   printf("ur string is: %s",str);
  return 0;
}
```

Strings

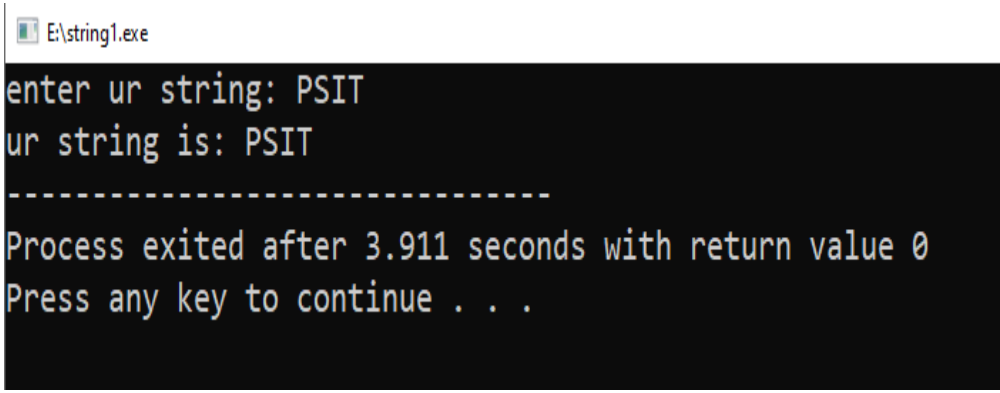
Taking string from user:

char str[20];

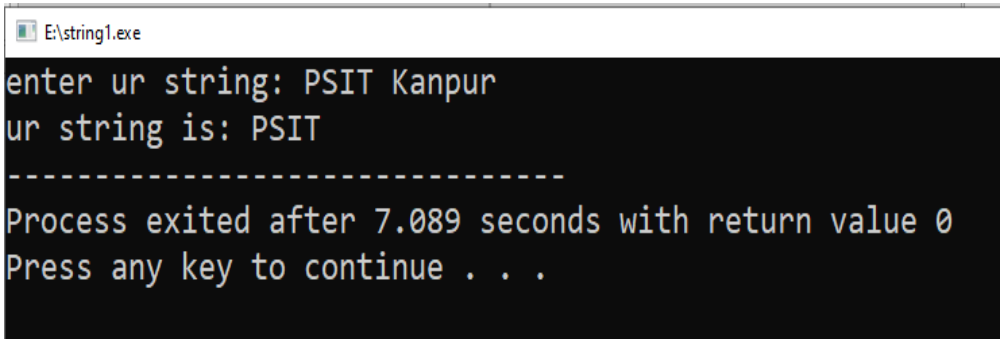
scanf("%s",str);

```
main()
{
   char str[20];
   printf("enter ur string: ");
   scanf("%s", str); // do not use   '&str' in scanf()

   printf("ur string is: %s",str);
  return 0;
}
```



scanf() can read only one word (string upto first white space)



Strings

Taking string from user:

char str[20];

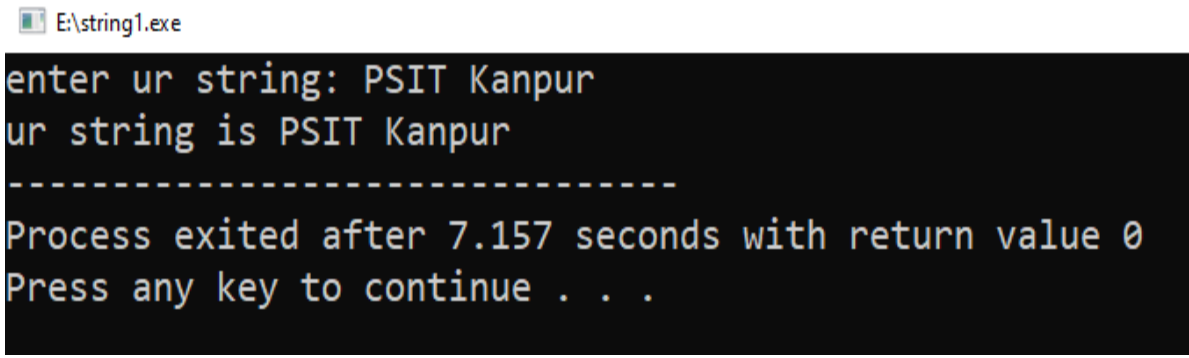
gets(str);

For printing string:

puts(str);

```
main()
{
   char str[20];
   printf(“enter ur string: ”);
   gets(str);

   printf("ur string is: %s",str);
  return 0;
}
```



String handling functions

1. *To find the length of a string :*
   strlen()
2. *To copy a string into other:*
   strcpy()
3. *To concatenate two strings:*
   strcat()
4. *To compare two strings:*
   strcmp()
5. *To reverse the string:*
   strrev()

Use header file:

string.h

Function	Use
strlen()	Finds length of a string
strcpy()	Copies a string into another
strcat()	Appends one string at the end of another
strcmp()	Compares two strings
strrev()	Reverses string

String handling functions

strlen( )

This function counts the number of characters present in a string.

syntax: **n=strlen(string_name);**

```
int main( )
   {
      char arr[30] = “PSIT,  Kanpur" ;
      int  len1, len2 ;
      len1 = strlen ( arr ) ;
      len2 = strlen ( "Humpty Dumpty" ) ;
      printf ( " %d  %d ", len1,len2 ) ;
      printf(“%d”,sizeof(arr)); //30
}
```

This function returns an integer, which is the length of string.

The output would be...

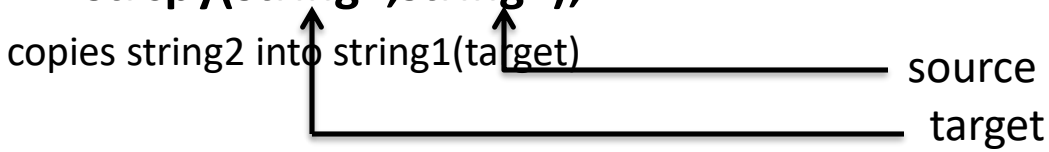
12 13

String handling functions

strcpy( )

This function copies the contents of one string into another.

Syntax: **strcpy(string1,string2);**




```
main( )
{
char st1[20 ] = “Rahul" ;
char st2[20] =“Abhishek”;
strcpy ( st1, st2 ) ;
puts(st1);
printf(“\n”);
puts(st2);
}
```

string1=string2 X

And here is the output...

Abhishek

Abhishek

String handling functions

strcat( )

This function concatenates a string at the end of the another string.

For example, "Rahul" and "Kumar" on concatenation would result into a string "RahulKumar".

Syntax: strcat(string1,string2);

this will append(add) string2 after string1.

```
main()
{
        char st1[20]="Rahul";
        char st2[20]="kumar";
         strcat(st1,st2);
        puts(st1);
        printf("\n");
        puts(st2);
}
```

```
RahulKumar
kumar
```

String handling functions

strcat( )

This function concatenates a string at the end of the another string.

For example, "Rahul" and "Kumar" on concatenation would result into a string "RahulKumar".

Syntax: strcat(string1,string2);

this will append(add) string2 after string1.

```
main()
 {
        char st1[20]="Rahul";
        char st2[20]="kumar";
       strcat(st1, "  " );
        strcat(st1,st2);
        puts(st1);
        printf("\n");
        puts(st2);
 }
```

```
Rahul Kumar
 kumar
```

String handling functions

strrev( )

This function reverse a string .

Syntax: strrev(string);

this will reverse the string.

```
main()
 {
        char st1[20]="Ramesh";
        char st2[20]="kumar";
      strrev(st1);
       strrev(st2);
        puts(st1);
        puts(st2);
 }
```

```
hsemaR
 ramuk
```

String handling functions

strcmp( )

This function is used to compare between two strings .

Syntax: strcmp(string1,string2);

if string1 and string 2 are same then it will return 0

```
main()
{
    int n;
    char st1[20]="Rahul";
      char st2[20]="Rahul";
    n=strcmp(st1,st2);
    printf("%d",n); //0
}
```

```
main()
{
    int n;
    char st1[20]="Rahul";
      char st2[20]="Ramesh";
    n=strcmp(st1,st2);
    printf("%d",n); //non zero value
}
```

String handling functions

strcmp( )

This function is used to compare between two strings .

Syntax: strcmp(string1,string2);

if string1 and string 2 are same then it will return 0

strcmp(string1,string2) =0 if string1=string2
=+1 if string1>string2
=-1 if string1<stiring2