strcpy

Copies one string to another

```
#include<stdio.h>
#include<string.h>
int main()
{
char text1[5],text2[5];
printf("Enter string1:");
gets(text1);
printf("\nEnter string2:");
gets(text2);
strcpy(text1,text2);
printf("\nString1:%s",text1);
printf("\nString2:%s",text2);
return 0;
}
```

```
Enter string1:hello

Enter string2:hi

String1:hi
String2:hi
```

strncpy

Copies n characters of one string to another

```
#include<stdio.h>
#include<string.h>
int main()
{
char text1[5],text2[5];
int n;
printf("Enter string1:");
gets(text1);
printf("\nEnter string2:");
gets(text2);
printf("\nEnter No of Characters:");
scanf("%d",&n);
strncpy(text1,text2,n);
printf("\nString1:%s",text1);
printf("\nString2:%s",text2);
return 0;
}
```

```
Enter string1:hello

Enter string2:howru

Enter No of
Characters:2

String1:hollo
String2:howru
```

strcmp

Compares two strings

```
#include<stdio.h>
#include<string.h>
int main()
{
char text1[5],text2[5];
int result;
printf("Enter string1:");
gets(text1);
printf("\nEnter string2:");
gets(text2);

result=strcmp(text1,text2);
```

```
Enter string1:hello
Enter string2:howru
String1 is less than
String2
```

```
Enter string1:hello
Enter string2:HELLO
String1 is greater than
String2
```

	if(result==0)
	printf("\nBoth strings are equal");
	else if(result>0)
	printf("\nString1 is greater than String2");
	else
	printf("\nString1 is less than String2");
	return 0;
	}
		Enter string1:hello
		Enter string2:hello
		Both strings are equal

strcmpi
 	
Compares two strings, non-case sensitive
	#include<stdio.h>
	#include<string.h>
	int main()
	{
	char text1[5],text2[5];
	int result;
	printf("Enter string1:");
	gets(text1);
	printf("\nEnter string2:");
	gets(text2);
	
	result=strcmpi(text1,text2);
	if(result==0)
	printf("\nBoth strings are equal");
	else if(result>0)
	printf("\nString1 is greater than String2");
	else
	printf("\nString1 is less than String2");
	return 0;
	}
		Enter string1:hello
		Enter string2:howru
		String1 is less than String2
		
		Enter string1:hello
		Enter string2:HELLO
		Both strings are equal
		
		Enter string1:hello
		Enter string2:hello
		Both strings are equal

strcat
 	
Appends a string
	#include<stdio.h>
	#include<string.h>
	int main()
	{
	char text1[10],text2[5];
	int length;
	printf("Enter string1:");
	gets(text1);
	printf("\nEnter string2:");
	gets(text2);
	strcat(text1,text2);
	printf("\nString1:%s",text1);
	printf("\nString2:%s",text2);
	return 0;
	}
		Enter string1:hello
		
		Enter string2:howru
		
		String1:hellohowru
		String2:howru

strncat
Appends n characters of string

```
#include<stdio.h>
#include<string.h>
int main()
{
char text1[10],text2[5];
int n;
printf("Enter string1:");
gets(text1);
printf("\nEnter string2:");
gets(text2);
printf("\nEnter no of characters to concate:");
scanf("%d",&n);
strncat(text1,text2,n);
printf("\nString1:%s",text1);
printf("\nString2:%s",text2);
return 0;
}
```

```
Enter string1:hello
Enter string2:howru

Enter no of characters
to concate:3

String1:hellohow
String2:howru
```

strncmp

Compares n characters of two strings

```
#include<stdio.h>
#include<string.h>
int main()
{
char text1[5],text2[5];
int result,n;
printf("Enter string1:");
gets(text1);
printf("\nEnter string2:");
gets(text2);
printf("\nEnter no of chars to compare from left:");
scanf("%d",&n);
result=strncmp(text1,text2,n);
if(result==0)
        printf("\nBoth strings are equal upto %d
characters",n);
else if(result>0)
        printf("\nString1 is greater than String2");
else
        printf("\nString1 is less than String2");
return 0;
}
```

```
Enter string1:hello
Enter string2:hell
Enter no of text to
compare from left:3
Both strings are equal
upto 3 characters
```

```
Enter string1:hello
Enter string2:HELL
Enter no of text to
compare from left:3
String1 is greater than
String2
```

strncmpi

Compares n characters of two strings ignoring cases.

```
#include<stdio.h>
#include<string.h>
int main()
{
char text1[5],text2[5];
int result,n;
printf("Enter string1:");
gets(text1);
printf("\nEnter string2:");
gets(text2);
printf("\nEnter no of text to compare from left:");
scanf("%d",&n);
result=strncmpi(text1,text2,n);
if(result==0)
    printf("\nBoth strings are equal upto %d characters",n);
else if(result>0)
    printf("\nString1 is greater than String2");
else
    printf("\nString1 is less than String2");
return 0;
}
```

```
Enter string1:hello
Enter string2:hell
Enter no of chars to compare from left:3
Both strings are equal upto 3 characters
```

```
Enter string1:hello
Enter string2:HELL
Enter no of chars to compare from left:3
Both strings are equal upto 3 characters
```

strrev

Reverses string

```
#include<stdio.h>
#include<string.h>
int main()
{
char text[5];
printf("Enter the string:");
gets(text);
printf("\nThe reverse of string is:%s",strrev(text));
return 0;
}
```

```
Enter the string:hello

The reverse of string is:olleh
```

strchr

Finds first occurrence of a given character

```
#include<stdio.h>
#include<string.h>
int main()
{
char text1[5],ch, *pos;
printf("Enter the string:");
gets(text1);
```

```
Enter the string:hello
Enter the character to search:l
```

	printf("Enter the character to search:");
	ch=getc(stdin);
	pos=strchr(text1,ch);
	if(pos!=0)
	printf("\nThe character:%c is at %d location inside the string:%s",ch,pos-text1+1,text1);
	else
	printf("\nThe character:%c is not inside the string:%s",ch,text1);
	return 0;
	}
		The character:l is at 3 location inside the string:hello

strrchr
 	
Finds last occurrence of given character in string
	#include<stdio.h>
	#include<string.h>
	int main()
	{
	char text1[5],ch, *pos;
	printf("Enter the string:");
	gets(text1);
	printf("Enter the character to search:");
	ch=getc(stdin);
	pos=strrchr(text1,ch);
	if(pos!=0)
	printf("\nThe %c is at %d (last occurance) inside the %s",ch,pos-text1+1,text1);
	else
	printf("\nThe %c is not inside the %s",ch,text1);
	return 0;
	}
		Enter the string:hello
		Enter the character to search:l
		
		The l is at 4 (last occurance) inside the hello

strstr
 	
Finds first occurrence of a given string
	#include<stdio.h>
	#include<string.h>
	int main()
	{
	char text1[5],text2[5], *pos;
	printf("Enter the string:");
	gets(text1);
	printf("Enter the string to search:");
	gets(text2);
	pos=strstr(text1,text2);
	if(pos!=0)
	printf("\nThe %s is at %d location inside the %s",text2,pos-text1+1,text1);
		Enter the string:hello
		Enter the string to search:lo
		
		The lo is at 4 location inside the hello

	else
	printf("\nThe %s is not inside the %s",text2,text1);
	return 0;
	}	

strlwr
 	
Converts a string to lowercase
	#include<stdio.h>
	#include<string.h>
	int main()
	{
	char text[5];
	clrscr();
	printf("Enter the string:");
	gets(text);
	printf("\nThe string in lower case is:%s",strlwr(text));
	return 0;
	}
		Enter the string:HeLLo
		
		The string in lower case is:hello

strupr
 	
Converts string to uppercase
	#include<stdio.h>
	#include<conio.h>
	#include<string.h>
	int main()
	{
	char text[5];
	int length;
	clrscr();
	printf("Enter the string:");
	gets(text);
	printf("\nThe string in Uppercase is:%s",strupr(text));
	getch();
	return 0;
	}
		Enter the string:HeLLo
		
		The string in Uppercase is:HELLO

5.8.2 In-built math functions

There are many math functions available in math.h and stdlib.h file. Some of the most used math functions are listed below.

Function	Description
floor ( )	This function returns the nearest integer which is less than or equal to the argument passed to this function.

round ( )	This function returns the nearest integer value of the float/double/long double argument passed to this function. If decimal value is from ".1 to .5", it returns integer value less than the argument. If decimal value is from ".6 to .9", it returns the integer value greater than the argument.
ceil ( )	This function returns nearest integer value which is greater than or equal to the argument passed to this function.
sin ( )	This function is used to calculate sine value.
cos ( )	This function is used to calculate cosine.
cosh ( )	This function is used to calculate hyperbolic cosine.
exp ( )	This function is used to calculate the exponential "e" to the x^{th} power.
tan ( )	This function is used to calculate tangent.
tanh ( )	This function is used to calculate hyperbolic tangent.
sinh ( )	This function is used to calculate hyperbolic sine.
log ( )	This function is used to calculates natural logarithm.
log10 ( )	This function is used to calculates base 10 logarithm.
sqrt ( )	This function is used to find square root of the argument passed to this function.
pow ( )	This is used to find the power of the given number.
trunc.(.)	This function truncates the decimal value from floating point value and returns integer value.

Example using math functions :

```
#include<stdio.h>
#include <math.h>
int main(){
printf("\n%f",ceil(3.6));
printf("\n%f",ceil(3.3));
printf("\n%f",floor(3.6));
printf("\n%f",floor(3.2));
printf("\n%f",sqrt(16));
printf("\n%f",sqrt(7));
printf("\n%f",pow(2,4));
printf("\n%f",pow(3,3));
printf("\n%d",abs(-12));
 return 0;
}
```

Output:

```
4.000000
4.000000
3.000000
3.000000
4.000000
2.645751
16.000000
27.000000
12
```

5.9 STORAGE CLASSES FOR VARIABLE

A storage class defines the scope (visibility) and life time of variables and/or functions within a C Program. There are four storage classes which can be used in a C program.

Storage Classes	Storage Place	Default Value	Scope	Lifetime
auto	RAM	Garbage Value	Local	Within function
extern	RAM	Zero	Global	Till the end of the main program May be declared anywhere in the program
static	RAM	Zero	Local	Till the end of the main program, Retains value between multiple functions call
register	Register	Garbage Value	Local	Within the function

Storage class tells the compiler how to store that variable.

Auto

Applied to local variables known to the function in which it is declared. Default is auto. Example:

```
auto int a;
```

Example

```
#include <stdio.h>
int main()
{
int a; //auto
char b;
float c;
printf("%d %c %f",a,b,c); // printing initial default value of automatic variables a, b, and c.
return 0;
}
```

Output:

```
garbage garbage garbage
```

Static

With Static storage local variables exist and retain its value even after the control is transferred to the calling function. Example,

```
static int x=30;
```

Example

```
#include<stdio.h>
void sum()
{
static int a = 10;
static int b = 24;
printf("%d %d \n",a,b);
a++;
b++;
}
void main()
{
int i;
for(i = 0; i< 3; i++)
{
sum(); // The static variables holds their value between multiple function calls.
}
}
```

Output:

```
10 24
11 25
12 26
```

Extern

Extern storage declares global variables known to all function in the file. For example,

```
extern int i;
```

Example 1 :

```
// filename: file1.c
int a;
int main(void)
{
  a = 2;
}
```

```
// filename: file2.c
// When this file is linked with file1.c, functions of this file can access a
extern int a;
int myfun()
{
  a = 2;
}
```

Register

Register storage is used for local variables which is stored in the register for fast access. Example

```
register int c;
```

Example 1

```
#include <stdio.h>
int main()
{
register int a; // variable a is allocated memory in the CPU register. The initial default value of a is 0.
printf("%d",a);
}
```

Output:

```
0
```

Example 2

```
#include <stdio.h>
int main()
{
register int a = 0;
printf("%u",&a); // This will give a compile time error since we can not access the address of a register variable.
}
```

Output:

```
main.c:5:5: error: address of register variable ?a? requested
printf("%u",&a);
^~~~~~
```