3. Develop program in C using library function

- **function** :- It is a piece of code/group of statements to complete certain operation. it has name for its identification.

→ functions are of two types -

1. User defined function :- The function developed by user to complete certain operation.
2. Predefined / Library function :-
   - → Library functions are inbuilt functions which are grouped together and placed in a common place called library.
   - → Each library function perform specific operation.
   - → Library functions are created by persons who designed C compiler.

- **Need of function** :-

→ If it is required to perform specific operation multiple times in a program, we have write same code again and again which is undesirable so, we can use function which perform that operation. function is an independent program that we are using in our main program, that is why function is also ~~known~~ referred as sub-program.

it has many advantages -

i) Program is divided into small functions (sub) which are easy to understand and debug if required.

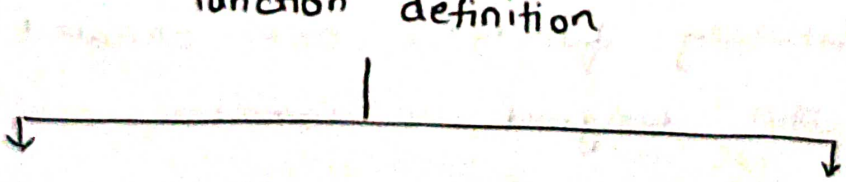
ii) Modification in program also become easier.

iii) Length of main program reduces

- **Elements of function :-**

There are some technical terms related to function

1) **function definition** :- function definition is giving name to a function and writing a piece of code inside { }. It has two main parts



a) function name
b) function type
c) list of parameters

d) Local variable declaration
e) function statement
f) Return statement

→ General form of function definition is

```
function_type   function_name (Parameter list)
{
    local variable declaration;
    executable statement 1;
```

executable statement 2 ;

- - - - - - - - - - - - -

return statement ;

}

→ function type specifies the type of value the function is expected to return to calling function.

e.g. - int , void , char

→ function name is a valid identifier & all rules of identifier are applicable to it . It must be unique.

→ Parameter list serves as input to the function that is given by calling function.

→ It is possible that function don't receive input and don't return back any value , then it can be represented as -

void function_name (void)

→ Local variable declaration in a function is done inside { } for executing specified task , but that variable remains valid only for that function

→ function statements are series of statements

that are used to perform specific task

→ Return statement returns the value evaluated by function, sometimes function don't return any value then it can be written as -

return 0 ;

2) Function call :-

→ A function can be called by writing as

function_name (actual parameters) ;

→ if we don't want to pass any parameter we can use void.

→ The function inside which another function is called is known as calling function & the function which is called is known as called function

3) Function declaration :-

→ Like variables, all functions in C program must be declared, before they are involved. A function declaration is done as -

function_type function_name (Parameter list) ;

→ These are declared outside main function.

Library function

The definition of library functions are present in their respective header file (having extension .h). So, to use these functions we need to include the header file in our program.

eg → To use printf() function, the header file <stdio.h> should be included like

```
#include <stdio.h>
```

① stdio.h library function :-

→ All c inbuilt functions which are declared in stdio.h header file are given below.

Printf(), scanf(), getc(), gets(), getchar(), puts(), Putchar() etc.

→ stdio stands for standard input output.

② conio.h library function :-

→ All c inbuilt functions which are declared in conio.h header file are given below.

clrscr(), getch(), getche() [it reads character from keyboard & echoes to output screen], textcolor() [This function is used to change the text color], textbackground() [This function

is used to change background of text

③ math.h library function :-

math.h file supports all mathematical related function in c language. All the arithmatic function used in c language are given below

function	Description
ⓐ floor()	This function returns the nearest integer which is less than or equal to argument passed to this function.
ⓑ round()	This function returns the nearest integer value of float/ double/ long argument passed to this function. If decimal value is from 0.1 to 0.5, it returns integer value less than argument. If decimal value is from 0.6 to 0.9, it return the integer value greate than argument.

syntax :

```
double floor(double x);
```

syntax :

```
double round (double a);
float roundf (float a);
long double roundl (long
                    double a);
```

ceil()
ntax:
ouble

ceil() This function returns the nearest integer value which is greater than or equal to argument passed to this function

ntax:
ouble ceil (double x);

sin() This function is used to calculate sine value.

Syntax - sin (i) // i is float type variable.

Similarly cos(), tan(), sinh(), cosh(), tanh(), exp(), log(), log10() are used in math function.

sqrt() To calculate square root of argument passed to this function

Pow() this function is used to find power of given number

syntax: pow (base, exponent)

trunc() This function truncates decimal value from floating point value & returns integer value.

syntax: double trunc (double a)

Example of math.h library function

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
int main()
{
    float i = 0.314;
    float j = 0.25;
    float k = 6.25;
    float sin_value = sin(i);
    float cos_value = cos(i);
    float tan_value = tan(i);
    float sinh_value = sinh(j);
    float cosh_value = cosh(j);
    float tanh_value = tanh(j);
    float log_value = log(k);
    float log10_value = log10(k);
    float exp_value = exp(k);
Printf ("The value of sin (%f) = %f \n", i, sin_value);
Printf ("The value of cos (%f) = %f \n", i, cos_value);
printf ("The value of tan (%f) = %f \n", i, tan_value);
Printf ("The value of sinh (%f) = %f \n", j, sinh_value);
Printf ("The value of cosh (%f) = %f \n", j, cosh_value);
Printf ("The value of tanh (%f) = %f \n", j, tanh_value);
Printf ("The value of log (%f) = %f \n", k, log_value);
Printf ("The value of log10 (%f) = %f \n", k, log10_value);
Printf ("The value of exp (%f) = %f \n", k, exp_value);
```

```
include <conio.h>
# include <stdio.h>
# include <math.h>
int main()
{
    Printf(" Sqrt of 16 = %.f \n", sqrt(16));
    Printf (" 2 power 4 = %.f \n", pow(2,4));
    Printf (" Truncated value of 16.7 = %.f \n", trunc(16.7));
    return 0;
}
```

Output :-

Sqrt of 16 = 4

2 power 4 = 16

Truncated value of 16.7 = 16.

(4) string.h library function :-

→ This file contain functions related to operation on string.. Following are inbuilt functions declared in string.h header file.

	string function	Description
(1)	strcat()	concates str2 at the end of string 1

Syntax - strcat (str1, str2); - str2 is concatenated at end of str1

→ As each string in C is ended up with a null character ('\0'). In strcat() operation null character of destination string is overwritten by source string's first character and null character will remain in destination string.

② strlen() gives length of string i.e no of integer in string. It does not count null character.

Syntax :-
i = strelen(string);

③ strcpy() Copies str2 in str1 - i.e Content of str2 get copied in string1

Syntax - strcpy(str1, str2); # ⊖

④ strcmp() Compares str2 characters with str1 characters one by one.

Syntax - strcmp(str1, str2);

for str1 & str2 being same, this function will return 0, for str1 character ASCII value being greater than str2 character ASCII value it will return positive number, otherwise negative number. This function is case sensitive

operations with

⑤ strrev() Reverses the given string ⑥

Syntax - strrev(string1);

⑥ — strdup() duplicates the string

⑦ strlwr() converts string to lower case

⑧ strupr() converts string to upper case

Example showing string.h functions

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
main()
{
    char s1[] = "Electrical";
    char s2[] = "engineering";
    char s3[30];
    clrscr();
    printf("\n The length of s1 is %d", strlen(s1));
    printf("\n The reversed string s1 is ");
    puts(strrev(s1));
    strcpy(s3, s1);
    puts(s3);
    strcmp(s1, s2);
    printf("\n strcmp for s1 & s2 returned %d",
```

strcmp(s1, s2));

```
    puts (strcat (s1, s2));
    getch();
}
```

User defined function

There are different ways to define user defined function -

i) Takes nothing, returns nothing
ii) Takes something, returns nothing
iii) Takes nothing, returns something
iv) Takes something, returns something

Suppose we want add() function, we will observe all four ways to define function -

(1) Taking nothing, returns nothing

Eg ⇒

```
main()
{   clrscr();
    add();
    getch();
}
add()
{   int a, b, c;
```

```
printf ("Enter two number :");
scanf ("%d %d", &a, &b);
c = a+b;
Printf ("Sum is %c", c);
}
```

→ clrscr() & getch() function are also not taking anything & not returning anything.

→ We should use void.

(11) Takes something returns nothing :-

→ A variable defined in one function cannot be called in other function, so if we want to supply data to a function this method is used.

```
void main()
{ int x, y;
  void add (int, int);
  clrscr();
  printf ("Enter two no.");
  scanf ("%d %d", &x, &y);
  add (x, y);
  getch();
}
```

```
void add (int a, int b)
{ int c;
  c= a+b;
  Printf (" Sum in %d", c);
}
```

→ We can use same name in main function (x, y) & add function for (a, b) variable. Separate memory space will be assigned to both.

(iii) Takes nothing, returns something :-

→ Now if we want to return the result of add() function to main function, we can use

```
return (c);
```

c is int type data, it will be returned to place where add() function was called so we have to assign it to some variable

```
s = add ();
```

→ By using return keyword we can return only one value and if we try to return more than one value like return (a,b,c) it will return only last value i.e c.

But return (a+b); is valid.

```
int   add (void);
void  main ()
{
   int  s;
   clrscr();
   s = add();
   printf  ("Sum is %d", s);
   getch();
}
    int  add()
{
    int   a,b,c;
    printf  ("Enter two numbers");
    scanf  ("%d %d", &a, &b);
     c= a+b;
     return  c;
}
```

Takes something return something

```
int   add (int , int);
void  main();
{
   int   s, x, y;
   clrscr();
   printf  ("Enter two no");
   scanf  ("%d %d", &x, &y);
```

```
    s = add (x, y);
    Printf ("Sum is %d", s);
    getch ();
}

int add (int a, int b)
{
    int c;
    c = a + b;
    return c;
}
```