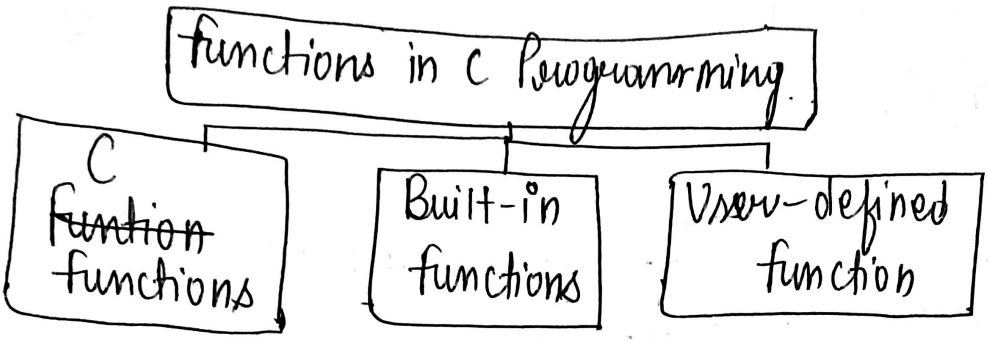
Unit → 3
lesson → 6 functions & need

Q-1 What is a function?
Ans. A function is a group of statements that together Perform a task. Every C Program has at least one function, which is main(), and all the most trivial programs can define additional functions.



- Advantages of a function?

① By using functions, we can avoid re writing same logic / code again and again in a program.
② We can call c functions any number of times in a Program and from any place in a program.
③ We can track a large C program easily when it is divided into multiple functions.
④ Reusability is the main achievement of C functions.
⑤ However, function calling is always a overhead in a C program.

- Types of functions.

① Library functions :- are the functions which are declared in the C header files such as Scanf(), printf(), gets(), puts(), ceil(), floor(), etc.

② User-defined functions :- are the functions which are created by the C programmer, so that he/she can use it many times. It reduces the Complexity of a big program and optimizes the Code.

- Function & Its need

```
int a =10 , b = 5, c ;
int product (int x, int y);          → function Prototype
int main (void)                      → Main fuction.
{
   c = product (a,b);                → function call
   Printf ("%i\n", c);
   return 0;
}
int product (int x, int y)           → fuction Definition
{
  return (x*y);
}
```

- Types of functions.

① Library functions:- are the functions which are declared in the C header files such as Scanf(), printf(), gets(), puts(), ceil(), floor(), etc.

② User-defined functions:- are the functions which are created by the C programmer, so that he/she can use it many times. It reduces the Complexity of a big program and optimizes the Code.

- Function & Its need

```
int a = 10, b = 5, c;
int product (int x, int y);          → function Prototype
int main (void)                      → Main fuction.
{
   c = product (a,b);                → function call
   Printf ("%d\n", c);
   return 0;
}
int product (int x, int y)           → fuction Definition
{
   return (x*y);
}
```

```
greatest (a, b, c);
}
Void greatest (int x, int y, int z)
{
int g;
If (a.b && a>c)
    g=a;
else if (b>a && b>c)
    g=b;
else
    g=c;
Printf ("greatest of 3 numbers = %d ; g);
}
```

- To print first 20 numbers & their square using functions.

```
# include<stdio.h>
Void Check ():
```

```
greatest (a, b, c);
}
Void greatest (int x, int y, int z)
{
int g;
If (a.b && a>c)
    g=a;
else if (b>a && b>c)
    g=b;
else
    g=c;
Printf ("greatest of 3 numbers = %d ; g);
}
```

- To print first 20 numbers & their square using functions.

```
# include<stdio.h>
Void Check ():
```

- To Check number is even/odd & Armstrong using function.

```
#include < stdio.h>
Void checkarmstrong (int);
void main ()
{ int n;
Printf("Enter number to check");
 Scanf ("%d", &n);
Check armstrong (n)
Checkeven (n);}
Void checkeven (int n)
 {
   If (n%2 ==0)
 Printf ("number is even");
      else
  Printf ("number is odd");
    }
Void checkarmstrong (int n)
   { int s=0, a=n, d;
  While (n!=0) {d=n %10; s=s+pow (d, 3); n=n/10;}
If (a == s) printf ('Armstrong');
else printf ("Not armstrong"); }
```

- To find sum, difference and product of 2 numbers.

```
int sum (int a, int b)
  {return a+b;}
int difference (int a, int b)
  {return a-b;}
```

```
int product (int a, int b)
{ return a * b; }
void main
{ int x, y, s;
Printf ("Enter 2 numbers");
Scanf ("%d %d", &a, &b);
S = Sum (x, y);
Printf ("Sum = %d", S);
S = difference (x, y);
Printf ("Difference = %d", S);
S = product (x, y);
Printf (" Product = %d ", S);
}
```

- Passing of arrays in fuction

There are 3 ways to declare the fuction which is intend to receive an array as an argument.

first way:-

Return_type function (type arrayname [])

Declaring black subscript notation [] is the widely used technique.

Second way:-

Return_type function (type array name [SIZE])

Optionally, we can define size in subscript notation [].

Third way:-

Return_type function (type * arrayname)

- Passing an array to function Example.

```
Int minarray (int arr[], int size) {
 int min = arr[0];
 int i=0;
for (i=1; i< size; i++) {
 If (min > arr[i] {
 min = arr[i];
 }
 } // end of for
  return min;
 } // end of function
  int main () {
  int i= 0, min = 0;
 int numbers [] = {4,5,7, 3,8,93}; //declaration of array
min = minarray (numbers; 6); // passing array with size.
 Printf ("minimum number is %d \n", min);
 return 0;
 }
```