Functions

Functions are subprograms. They perform well defined tasks. They may be developed separately and tested. In other words, a function is a self-contained block of statements that perform a coherent(relational) task of some kind. Every c program can be thought of as a collection of these functions.

It is something like hiring a person to do a specific job for you. Sometimes the interaction with this person is very simple sometimes it’s complex.

Functions can be used in four different ways

1.No return and no parameter passing

Void add ();

2. Return value and passing parameter.

Int add (int x, int y)

3. Return value and no parameter passing.

Int add ()

4. No return and passing parameter

void add (int x, int y)

Syntax of function

Return type function name (parameters/ method of passing parameters)

1.No return and no parameter passing

1. Write a program to print sum, product and quotient of two numbers 32 and 8. Using no return and no parameter passing

```
#include<stdio.h>
#include<conio.h>
void  sum ();    //declaration
 void main()
      {
       clrscr();
       sum();
        getch();
        }
       void sum()
       {
        int x,y,s,p,q;
       x=32;
       y =8;
       s=x+y;
       p=x*y;
       q=x/y;
       printf("sum=%d",s);
       printf("\n product=%d",p);
       printf("\n quotient=\t%d",q);
       }
```

Only call/ main program

function definition

3 *Write algorithm, flow chart and program to input age of person and print in days with a appropriate format.* //No return no argument

```
#include<stdio.h>
#include<conio.h>
void year();
        void main()
         {
         clrscr();
          year();
         getch();
          }
          void year()
          {
int year,month,day,days;
 printf("Enter your age in year");
 scanf("%d",&year);
 printf("Enter month");
 scanf("%d",&month);
 printf("Enter day");
 scanf("%d",&day);
 days=(year*365)+(month*30)+day;
 printf("days=%d",days);
 }
```

WAP to read input number is negative or positive No return and no parameter passing

```
#include <stdio.h>
 #include <conio.h>
 void checknum();
 void main()
 {
 clrscr();
 checknum();
 getch();
 }
 void checknum()
 {
 int n;
 printf("enter the num: ");
 scanf("%d",&n);
 if(n>0)
 printf("positive");
 else
 printf("Negetive");
}
```

WAP to print given number is negative or positive using function no return and no parameter passing

```
#include <stdio.h>
 #include <conio.h>
 void checknum();
 void main()
 {
 checknum();

 clrscr();
 getch();
 }
 void checknum()
 {
 int n;
 printf("enter the num: ");
 scanf("%d",&n);
 if(n>0)
 printf("positive");
 else
 printf("Negetive");
}
```

Write a program to input length & breadth of a room and calculate and print its area and perimeter. no return and no parameter passing

```
#include<stdio.h>
#include<conio.h>
void area();
   void main()
       {
       clrscr();
       area();
       getch();
       }
       void area()
       {
       int l,b,A,P;
       printf("Enter the L:");
       scanf("%d",&l);
       printf("Enter the B:");
       scanf("%d",&b);
       A=l*b;
       P=2*(l+b);
      printf("area=%d",A);
      printf("\nperimeter=%d",P);
       }
```

2. Return value and passing parameter.

Int add (int x, int y)

WAP to print area where l=30 and b=10 return value and passing parameter.

```
#include <stdio.h>
#include <conio.h>
  int area(int x, int y); //return value with argument/parameter
  //{
  void main()
{
   clrscr();
  int l,b,c;
  l=30;
  b=10;
  c=area(l, b);
  printf("area=%d",c);
  getch();
   }
 int area( int x, int y)
 {
  int a;
  a=x*y;
  return a;
  }
```

Write a program to read base and altitude of a triangle and prints its area (return value with argument)

```
##include<stdio.h>
#include<conio.h>     //return value  with  a gument
int  area ( int x, int y );
   void main()
            {
        int h,b,v;
        clrscr();
        printf("Enter the Value of Altitude of a triangle  H= ");
        scanf("%d",&h);
        printf("Enter the Value of Base of a triangle  B= ");
        scanf("%d",&b);
        v = area(h,b);
        printf("Area=  %d",v);
        getch();
        }
        int area( int x,  int y)
        {
        int a;
        a=(x*y)/2;
       return a ;
       }
```

```
                                                    //Find out the input is factorial
#include<stdio.h>
#include<conio.h>
int fact (int);
 void main()
    {
    int n,factorial;
   printf("enter the number=");
   scanf("%d",&n);
   factorial=fact(n);
   printf("factorial=%d",factorial);
   getch();
   }
    int fact (int x)
    {
    int f=1,i;
    for (i=x;i>=1;i--)

   f=f*i;
   return(f);
    }
```

WAP total amount of p,n,r simple interest per annum using return value with passing parameter

```
include<stdio.h>
#include<conio.h
float interest( float x,float y,float z);
  void main()
    {
    clrscr();
       float I,P,T,R,s;
       printf("Enter the Value of P = ");
       scanf("%f",&P);
       printf("Enter the Value of T= ");
       scanf("%f",&T);
       printf("Enter the Value of R= ");
       scanf("%f",&R);
       s=interest(P,T,R);
       printf("Interest=  %.2f",s);
      getch();
       }
```

```
float interest(float x, float y, float z)
        {
        float I;
         I=(x*y*z)/100;
        return I;
        }
```

3. Return value and no parameter passing.

Int add ()

/*WAP total amount of p,n,r simple interest per annum using Return value no argument*/

```
#include<stdio.h>
#include<conio.h>
float interest();
 void main()
 {
 int s;
 clrscr();
  s=interest();
 printf("Interest=  %.2f",s);
  getch();
  }
```

```
float interest()
      {
      clrscr();
      float I,P,T,R;
      printf("Enter the Value of P = ");
      scanf("%f",&P);
      printf("Enter the Value of T= ");
      scanf("%f",&T);
      printf("Enter the Value of R= ");
      scanf("%f",&R);
      I=(P*T*R)/100;
      return I;
      }
```

Write program to input 5 digit integer and print sum of digit in it.

```
#include<stdio.h>
#include<conio.h>
int arm ();
void main()
  {
       clrscr();
       int s;
       s=arm();
       printf( "arm=%d",s);
       getch();
       }
       int arm ()
       {
int sum,d1,d2,d3,d4,d5,n,n1,n2,n3,n4,n5;
       printf("Enter the Value of Five Digit = ");
       scanf("%d",&n);
       d1=n%10;
       n1=n/10;
       d2=n1%10;
       n2=n1/10;
       d3=n2%10;
       n3=n2/10;
       d4=n3%10;
       n4=n3/10;
       d5=n4%10;
       sum=d1+d2+d3+d4+d5;
       return sum;
       }
```

Write a program to read base and altitude of a triangle and prints its area (return value no argument)

```
#include<stdio.h>
#include<conio.h>      //return value no argument
float area();
   void main()
                    {
       float V,A;
       clrscr();
       V=area();
       printf("Area=  %.1f",V);
       getch();
       }
       float area()
       {
        float A,B,H,a;
              printf("Enter the Value of Altitude of a
triangle  H= ");
              scanf("%f",&H);
              printf("Enter the Value of Base of a
triangle  B= ");
              scanf("%f",&B);
              A=(B*H)/2;
             return A ;

              }
```

4. No return and passing parameter

void add (int x, int y)

Write program to input 3 digit integer and print sum of digit in it.

//no rutern value with passing parameter

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void checkarm(int n);
 void main()
 {
 int num;
 printf("enter the arm: ");
 scanf("%d",&num);
 checkarm(num);
 getch();
 }
 void checkarm(int n)
 {
 int n1,d1,d2,d3,arm;
 n1=n;
 d1=n%10;
 n=n/10;
 d2=n%10;
 n=n/10;
 d3=n;
 arm=pow(d1,3)+pow(d2,3)+pow(d3,3);
 if (n1==arm)
 printf("armstrong");
 else
 printf("No armstrong");
 }
```

WAP to print given number is negative or positive using function no return and with parameter passing

Write a program to find out whether the given 4 digit number (year) is a leap year. Using no return passing par.

```
#include <stdio.h>
#include <conio.h>
void year(int);
void main()
{
int y;
printf("enter the year=");
scanf("%d",&y);
year(y);
getch();
 }
void year(int y)
{
if((y%4==0&&y%100!=0)||y%400==0)
printf("leap year");
else
printf("NOt leap");
}
```

/*Program for printing factorial for input number using for loop*/

```
#include<stdio.h>
#include<conio.h>
 void main()
   {
   int n,i,fact=1;
 printf("enter the number=");
 scanf("%d",&n);
 for (i= 1;i<=n;i++)
 fact=fact*i;
   printf("fact=%d",fact);
   getch();
   }
```

/*WAP to find the factorial of a number using recursiv function*/

```
#include<stdio.h>
#include<conio.h>
int fact (int);
 void main()
   {
   int n,factorial;
  printf("enter the number=");
  scanf("%d",&n);
  factorial=fact(n);
  printf("factorial=%d",factorial);
  getch();
  }
   int fact (int x)
   {
   int f=1,i;
   for (i=x;i>=1;i--)

  f=f*i;
  return(f);
   }
```

/*WAP to find the factorial of a number using recursiv function*/

```
#include<stdio.h>
#include<conio.h>
int fact(int n);
void main()
    {
   int n;
   clrscr();
   printf("Enter any (n)");
   scanf("%d",&n);
   printf("fact=%d",fact(n));
   getch();
   }
   int fact(int n)
   {
   if(n<=1)
   return(1);
   else
   return(n*fact(n-1));
    }
```

Recursion basics - using factorial - .flv

- 5!
- 5*4
- 5*4*3
- 5*4*3*2
- 5*4*3*2*1!
- 5*4*3*2*1

Compare the two sets

```
Main()
{
Int  i=1;
i++;
}
```

```
Int i=1
Main  ()
{
i++;
}
```

Local variable

These variable only exist inside the specific function. They are unknown to other functions and to the main program. Local variables cease to exit once the function that created the is completed. They are recreated each time a function is executed.

```
Main()
{
Int  i=1;
i++;
}
```

//Program to swap value of variables using local variable

```
#include <stdio.h>
#include <conio.h>
void swap( int a, int b);
void main()
{
int a,b;
printf( "enter the value of a=");
scanf("%d",&a);
printf( "enter the value of b=");
scanf("%d",&b);
printf("\n before SWAP\n");
printf("a=%d\n",a);
printf("b=%d\n",b);
swap(a,b);
getch();
}
void swap(int a,int b)
{
int temp;
temp=a;
a=b;
b=temp;
printf( "After SWAP");
printf("\na=%d\n",a);
printf("b=%d\n",b);
}
```

Global variables

These variables can be used by any function in the program. They are implemented by associating memory location with variable names. They do not get recreated if the function is called .

```
Int n=1
Main ()
{
i++;
}
```

//Program to swap value of variables using global variable using function no return and no parameter passing

```
#include <stdio.h>
#include <conio.h>
int a,b;
void swap();
void main()
{
printf( "enter the value of a=");
scanf("%d",&a);
printf( "enter the value of b=");
scanf("%d",&b);
printf("\n before SWAP\n");
printf("a=%d\n",a);
printf("b=%d\n",b);
swap();
getch();
}
void swap()
{
int temp;
temp=a;
a=b;
b=temp;
printf( "After SWAP");
printf("\na=%d\n",a);
printf("b=%d\n",b);
}
```