Recursive Function

Dr. Rahul Das Gupta

Recursive Function Definition

A **recursive function** is a **function** that calls itself during its execution until some **terminating condition** is reached.

Example: fact(n)=n*fact(n-1)   if n>0
fact(0)=1

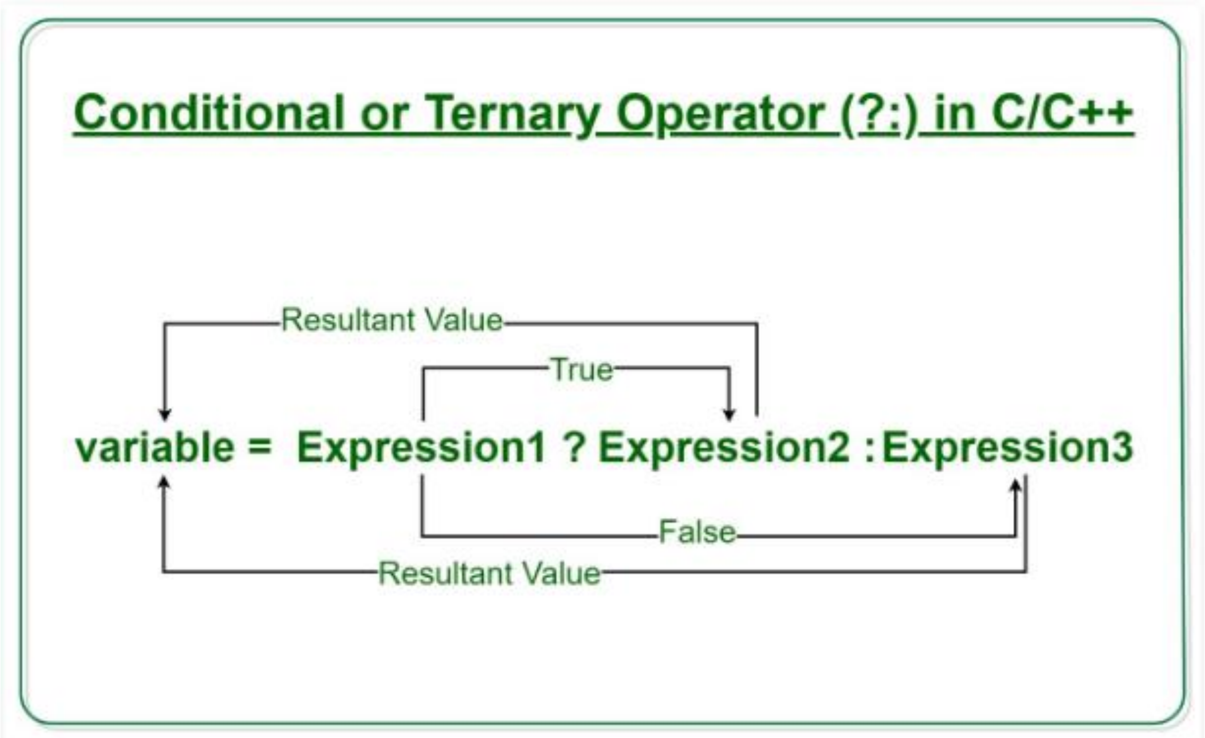
```
int  factorial (int n)
{
    if (n==0)      /* terminating condition */
       return (1);
    else
       return (n*factorial(n-1));     /* recursive call  */
}
```

The above example can be alternatively written as follows using **Conditional operator or Ternary operator**

```
int  fact( int x)
{
    return((x= =0)?  1 : x*fact(x-1));
}
```

Conditional operator or Ternary operator

The conditional operator is kind of similar to the **if-else statement** as it does follow the same algorithm as of **if-else statement** but the conditional operator takes less space and helps to write the if-else statements in the shortest way possible.


Conditional or Ternary Operator (?:) in C/C++
Resultant Value
True
variable = Expression1 ? Expression2 : Expression3
False
Resultant Value

A Menu Based C-Program for the determination of the following using recurssion
(i) factorial of an integer
(ii)GCD
(iii)Combination
(iv)Fibonacci Number generation

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

int   fact( int );
int    gcd( int, int );
int    Combination( int, int );
int    fibo( int );
void   menu ( );

void  main()
{
   clrscr();
   menu();
}
```

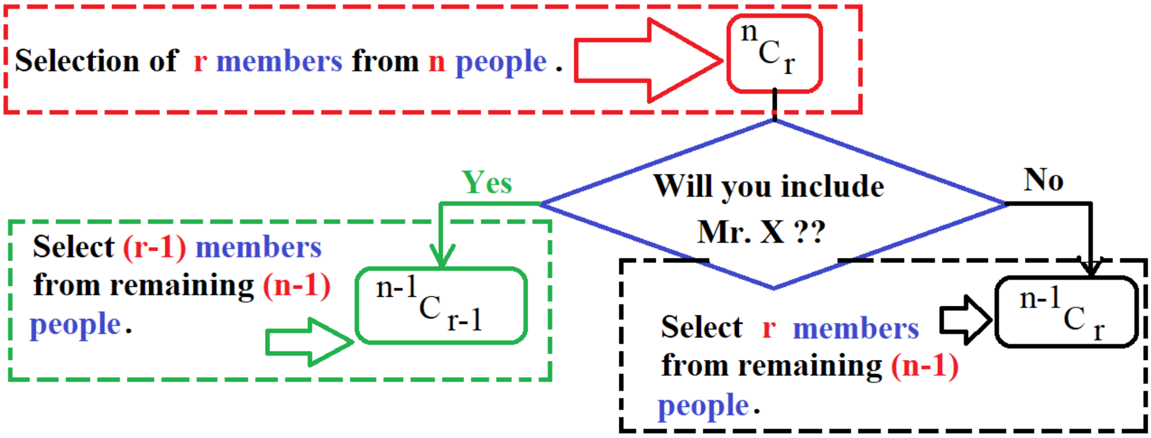
```
int  fact( int x)
{
     return((x==0)?  1 : x*fact(x-1));
}
/*
GCD(x,0) = GCD(0,x) = x     [ Here, x=x*1 and 0=0*x. Hence, the highest
common factor (HCF) between x and 0 is x.]
Euclidean Algorithm:
X|  Y |
  -------
   Y%X| X
GCD(X,Y) = GCD(Y%X, X)
Case when X>Y:
5 | 2  | 0     2=0*5+2
     2  | 5
*/

int   gcd( int  x, int y)
{
     if(x==0)
               return(y);
     else if (y==0)
               return(x);
     else
            return(gcd(y%x, x));
}
/*==================================================
```

Binomial Coefficients or Combinational Coefficients

${}^{n}C_{r} = 1 \quad (\text{if } r = 0 \text{ or } r = n)$

${}^{n}C_{r} = {}^{n-1}C_{r-1} + {}^{n-1}C_{r} \quad (\text{if } 0<r<n)$



```
=============================================================*/
int   Combination( int  n, int  r )
{
      if(r ==0 || r  ==n)
            return(1);
     else
            return(Combination(n-1, r-1) + Combination(n-1, r ));
}


/* Fibonacci Series:  0, 1, 1, 2, ….
fibo (n) = fibo (n-2) + fibo (n-1)  [when n>=2]
             = n [when n<=1]
fibo (0) = 0 and fibo (1) = 1
*/

int   fibo( int n)
{
    if(n<=1)
              return(n);
    else
              return (fibo(n-2) + fibo(n-1));
}
```

```
void  menu ( )
{
  int n, r, i, x, y;
  int choice;
  char  ans;
   do
   {
     printf(“\n    MENU
               \n 1: for Factorial.
               \n 2: for GCD.
               \n 3:  for Fibonacci Series.
               \n 4:  for Combination Coefficient.
               \n 5:  for EXIT.
               \n  Enter your choice (1 to 5) :”);
     fflush(stdin);
     scanf(“%d”, &choice);
     switch(choice)
     {
        case 1:
                 printf(“ \n  Enter the integer :”);
                 fflush(stdin);
                 scanf(“%d”,&n);
                 printf(“ \n  Result %d! = %d ”, n, fact(n));
                 break;
        case 2:
                 printf(“ \n  Enter the first integer :”);
                 fflush(stdin);
                 scanf(“%d”,&x);
                 printf(“ \n  Enter the second integer :”);
                 fflush(stdin);
                 scanf(“%d”,&y);
                 printf(“ \n  Result: GCD (%d,%d) = %d ”, x, y, gcd (x,y));
                 break;
        case 3:
                 printf(“ \n  Enter the number of terms :”);
                 fflush(stdin);
                 scanf(“%d”,&n);
```

```
            for(i=0;i<n;i++)
                printf(“ \t %d”, fibo(i));
            break;
      case 4:
            printf(“ \n  Enter the value of n:”);
            fflush(stdin);
            scanf(“%d”, &n);
            printf(“ \n  Enter the value of r :”);
            fflush(stdin);
            scanf(“%d”, &r);
            printf(“ \n  Result: C(%d,%d) = %d ”, n, r, Combination (n,r));
            break;

      case 5:
           exit(1);
      default:
            printf(“ \n Wrong choice…”);
    }
   printf(“\n Do you want to continue (Y/N) ?”);
   fflush(stdin);
   ans=getchar( );
  }while(ans == ‘Y’ || ans ==’y’);


}
```