Recursion:-

Recursion is the technique of making a function call itself. This technique provides a way to break complicated problems down into simple problems which are easier to solve. Recursion is the process of repeating items in a self-similar way. In programming languages, if a program allows you to call a function inside the same function, then it is called a recursive call of the function.

```
void recursion() {
  recursion(); /* function calls itself */
}

int main() {
  recursion();
}
```

Example:

```
#include <stdio.h>
int fact (int);
int main()
{
   int n,f;
   printf("Enter the number whose factorial you want to calculate?");
   scanf("%d",&n);
   f = fact(n);
   printf("factorial = %d",f);
}
int fact(int n)
{
   if (n==0)
   {
      return 0;
   }
   else if ( n == 1)
   {
      return 1;
   }
   else
   {
      return n*fact(n-1);
   }
}
```

Output

```
Enter the number whose factorial you want to calculate?5
factorial = 120
```

Example of recursion in C

Let's see an example to find the nth term of the Fibonacci series.

```
#include<stdio.h>

int fibonacci(int);

void main ()

{

   int n,f;

   printf("Enter the value of n?");

   scanf("%d",&n);

   f = fibonacci(n);

   printf("%d",f);

}

int fibonacci (int n)

{

   if (n==0)

   {

   return 0;

   }

   else if (n == 1)

   {

      return 1;
```

```
    }

    else

    {

        return fibonacci(n-1)+fibonacci(n-2);

    }

}
```

Output

```
Enter the value of n?12
144
```

Example: Sum of Natural Numbers Using Recursion

```
#include <stdio.h>
int sum(int n);

int main() {
   int number, result;

   printf("Enter a positive integer: ");
   scanf("%d", &number);

   result = sum(number);

   printf("sum = %d", result);
   return 0;
}

int sum(int n) {
   if (n != 0)
      // sum() function calls itself
      return n + sum(n-1);
   else
      return n;
}
```

Output

Enter a positive integer:3
sum = 6

Explanation of above example:-