Application of Stack

Recursion:

The process in which a function calls itself directly or indirectly is called recursion.

Types of recursion

1. **Direct Recursion**: These can be further categorized into four types:

a) **Tail Recursion**: If a recursive function calling itself and that recursive call is the last statement in the function then it's known as Tail Recursion. After that call the recursive function performs nothing. The function has to process or perform any operation at the time of calling and it does nothing at returning time.

b) **Head Recursion**: If a recursive function calling itself and that recursive call is the first statement in the function then it's known as Head Recursion.There's no statement, no operation before the call. The function doesn't have to process or perform any operation at the time of calling and all operations are done at returning time.

c) **Tree Recursion**: To understand Tree Recursion let's first understand Linear Recursion. If a recursive function calling itself for one time then it's known as Linear Recursion. Otherwise if a recursive function calling itself for more than one time then it's known as Tree Recursion.

d) **Nested Recursion**: In this recursion, a recursive function will pass the parameter as a recursive call.That means "recursion inside recursion".

2. **Indirect Recursion**: In this recursion, there may be more than one functions and they are calling one another in a circular manner.

Programs

Write a program in C to print Fibonacci series

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

```
int Fibonacci(int);

void main()

{

  int n, i = 0, c;

   scanf("%d",&n);

   printf("Fibonacci series\n");

for ( c = 1 ; c <= n ; c++ )

  {

  printf("%d\n", Fibonacci(i));

   i++;

 }

 return 0;

}

int Fibonacci(int n)

{

  if ( n == 0 )

 return 0;

 else if ( n == 1 )

  return 1;

 else
```

```
 return ( Fibonacci(n-1) + Fibonacci(n-2) );

}
```