Types of Recursion

Direct Recursion

A function is directly recursive if it explicitly calls itself.

```
int Func(int n) {
   if(n == 0)
     return n;
   else
     return(Func(n - 1));
}
```

Indirect Recursion

A function is indirectly recursive if it contains a call to another function which ultimately calls it.

```
int Func1(int n) {
   if(n == 0)
     return n;
   else
     return(Func2(n));
}
int Func2(int x) {
   return Func1(x - 1);
}
```

Tail Recursion

A recursive function is tail recursive if no operations are pending to be performed when the recursive function returns to its caller.

/* Non-tail recursion */

```
int Fact(int n) {
   return Fact1(n, 1);
   }
int Fact2(int n, int res) {
   if(n == 1)
     return res;
   else
     return(n - 1, n * res);
}
```

/* Tail recursion */

```
int Fact(int n) {
   if(n == 1)
     return 1;
   else
     return(n * Fact(n - 1));
}
```

Points to Remember

• A stack is a linear data structure in which elements are added and removed only from one end, which is called the top. Hence, a stack is called a LIFO (Last-In, First-Out) data structure as the element that is inserted last is the first one to be taken out.

• In the computer's memory, stacks can be implemented using either linked list or single arrays.

• The storage requirment or linked representation of stacks with n elements is $O(n)$ and the typical time requirement for operations is $O(1)$.

• Infix, prefix, and postfix notations are different but equivalent notations of writing algebraic expressions.

• In postfix notation, operators are placed after the operands, whereas in prefix notation, operators are placed before the operands.

• Postfix notations are evaluated using stacks. Every character of the postfix expression is scanned from left to right. If the character is an operand, it is pushed onto the stack. Else, if it is an operator, then the top two values are popped from the stack and the operator is applied on these values. The result is then pushedonto the stack.

• Multiple stacks means to have more than one stack in the same array of sufficient size.

• A recursive function is defined as a function that calls itself to solve a smaller version of its task until a final call is made which does not require a call to itself. They are implemented using system stack.