Recursion

Ex. 1

Write a recursive definition of the multiply function, then implement it.

int multiply(int a, int b) ;

Ex. 2

Write a recursive definition of a function which displays numbers from 1 to 30. Implement it.

Ex. 3

We want to write a solution to the tower of Hanoï problem defined as follows :

1. 3 pegs « A », « B » and « C »
2. N disks of differing diameters are placed on peg « A » so that a larger disk is always below a smaller one

Goal : Move all disks to « C » using « B » as an auxiliary

Constraints :

1. *Only the top disk on any peg can be moved*
2. *A larger disk may never rest on a smaller one*
3.

Implement the following function which displays the movements of disks betwen pegs

void towers(int n, char frompeg, char topeg, char auxpeg)

Ex. 4

Consider the following definition of algebraic expressions :

- An **expression** is a *term* followed by a « + » sign followed by a *term*, or a *term* alone
- A **term** is a *factor* followed by an asterisk followed by a *factor*, or a *factor* alone
- A **factor** is either a *letter* or an *expression* enclosed in parentheses

Write a program that reads a character string and then prints « valid » or « invalid » based on the previous definition

- int getsymb(char *str, int length, int *ppos)
- int expr(char *str, int length, int *ppos)
- int term(char *str, int length, int *ppos)
- int factor(char *str, int length, int *ppos)

Ex. 5 (Bonus)

Give a recursive function definition that transforms a prefix expression into a postfix expression.

Prefix : -++A*BCD*EF

Postfix : ABC*+D+EF*-