CISC 3321

Programming Assignment 8 (20 Pts)

Due at 10:00 PM Central (MyCampus)

Topic(s): Recursion & Method

Solving the Tower of Belton problem:

Tower of Belton (Hanoi)

Of all the computer science's classical problems, Tower of Hanoi (above) seems to be the most intriguing and every computer scientist must wrestle with it. Legend has it that in a temple in the Far East, priests are attempting to move a stack of disks from one peg to another. The initial stack has 64 disks threaded on one peg and arranged from bottom to top by decreasing size. The priests are attempting to move the stack from this peg to a second peg under the constraints that exactly one disk is moved at a time and at no time may a larger disk be placed above a smaller disk. A third peg is available for temporary holding disks. Supposedly, the world will end when the priests complete their task, so there was really no incentive to rush.

If the legend were true, and if the priests were able to move disks at a rate of one per second, using the smallest number of moves, it would take them $2^{64}-1$ seconds or roughly 585 billion years or 18,446,744,073,709,551,615 turns to finish, or about 127 times the current age of the sun.

Let's assume that the priests are attempting to move the disks from peg 1 to peg 3. We wish to develop an algorithm that will display the precise sequence of peg-to-peg disks transfers. Approaching this problem using the conventional (iterative) methods is obviously a daunting, if not impossible, task. However, if we think of this problem recursively, it immediately becomes tractable. This means we can now view moving *n* disks in terms of moving *n-1* disks (recursion) as follows:

a) Move *n-1* disks from peg 1 to peg 2, using peg 3 as a temporary holding area.
b) Move the last disk (the largest) from peg 1 to peg 3.
c) Move the *n-1* disks from peg 2 to peg 3, using peg 1 as a temporary holding area.

The process ends when the last task involves moving *n = 1* disk (i.e., the base case), which is accomplished by simply moving the disk, without the need for a temporary holding area.

You are to write a program to solve the Tower of Belton (very similar to that of Hanoi) problem with the following specs:
Allow the user to enter the number of disks and use the recursive tower method with four parameters:

- the number of disks to be moved
- the peg on which these disks are initially threaded
- the peg to which this stack of disks is to be moved, and
- the peg to be used as a temporary holding area

Your program should display the precise instructions it will take to move the disks from the starting peg to the destination peg. For instance, to move a stack of three disks from peg 1 to peg 3, your application should display the following series of moves:

1 - - > 3 (meaning moving one disk from peg 1 to peg 3)
1 - - > 2
3 - - > 2
1 - - > 3
2 - - > 1
2 - - > 3
1 - - > 3