CDA 3103 – Project 2 (Assembly Language)

Programming Assignment: [Due 6/30]

Work in group of two people (maximum) and submit only one assignment per group. Use the SPIM simulator for coding the assignment [http://pages.cs.wisc.edu/~larus/spim.html]. For the given problem you can have the visual game play at (http://www.mathsisfun.com/games/towerofhanoi.html). You can find a good tutorial for Towers of Hanoi at (http://www.cems.uvm.edu/~rsnapp/teaching/cs32/lectures/hanoi.pdf)

Problem A.10 [page A-83]

Using SPIM, write and test a recursive program for solving the classic mathematical recreation, the Towers of Hanoi puzzle. (This will require the use of stack frames to support recursion.) The puzzle consists of three pegs (1, 2, and 3) and $n$ disks (the number $n$ can vary; typical values might be in the range from 1 to 8). Disk 1 is smaller than disk 2, which is in turn smaller than disk 3, and so forth, with disk $n$ being the largest. Initially, all the disks are on peg 1, starting with disk $n$ on the bottom, disk $n-1$ on top of that, and so forth, up to disk 1 on the top. The goal is to move all the disks to peg 2. You may only move one disk at a time, that is, the top disk from any of the three pegs onto the top of either of the other two pegs. Moreover, there is a constraint: You must not place a larger disk on top of a smaller disk.

The C program below can be used to help write your assembly language program.

(Hint: See the SPIM system calls on pages A-43 and A-45)

```c
/* move n smallest disks from start to finish using extra */
void hanoi(int n, int start, int finish, int extra){
    if(n != 0){
        hanoi(n-1, start, extra, finish);
        print_string("Move disk");
        print_int(n);
        print_string("from peg");
        print_int(start);
        print_string("to peg");
        print_int(finish);
        print_string(".
")
        hanoi(n-1, extra, finish, start);
    }
}
main(){
    int n;
    print_string("Enter number of disks>");
    n = read_int();
    hanoi(n, 1, 2, 3);
    return 0;
}
```
Grading Guidelines
For your information here are the grading guidelines for the SPIM component of the assignment. Submit the code files in .asm extension before deadline

• Program compiles without errors (and appears to address the problem) : 50pts
• Program executes correctly: (additional) 40 points
• Documentation and description of the program: 10 points

For the comments try to comments it generally for someone who is not aware of the assignment can understand what the program is computing