

*Recursion - Towers of Hanoi***Introduction:**

Write a MIPS assembly program to solve the Towers of Hanoi Puzzle. Implement the recursive algorithm that was introduced in your CSCI 160 course.

The recursive Hanoi function will receive its four 32-bit integer arguments pushed onto the stack in the order listed in the Java version below. The recursive Hanoi function must allocate space on the stack to preserve the contents of \$ra and any other registers that your recursive function uses as local variables. The recursive function must end by restoring saved registers and releasing allocated stack space and jumping to the return address.

Your solution must implement the stack-based recursive function call mechanism. Do not use loops and branches to simulate the effect of recursive function calls. This will be extra credit.

**Grading Criteria**

Declaration of data constants and strings	5
Prompt message and input of number of discs	5
Main pushes parameters for initial call to Hanoi	5
Preserve register values \$sp and locals	10
Access of pushed parameters from stack	10
Recursive base case	10
Push parameters for recursive call	10
Recursive call	10
Use of saved parameter values from stack	10
Restore saved registers	10
Release stack space	5
Comments	10
Total	100

**Deliverables:**

Submit the source file Towers.asm by upload to Brightspace. Do not zip it!

The CSCI 160 Towers of Hanoi algorithm is presented below.

```
import java.util.Scanner;

public class Towers{
    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);

        System.out.println("Enter number of disks: ");
        int numDisks = scan.nextInt();

        System.out.println("Start with " + numDisks + " on peg A.");
        hanoi(numDisks, 'A', 'C', 'B');
    }

    public static void hanoi(int n, char source, char dest, char temp){
        if(n == 1){
            System.out.println("Move disk from " + source + " to " + dest);
        }
        else{
            hanoi(n - 1, source, temp, dest);
            System.out.println("Move disk from " + source + " to " + dest);
            hanoi(n - 1, temp, dest, source);
        }
    }
}
```