Assignment 4

(Due Date April 5th 16:00 est)

Part A.

Towers of Hanoi (ToH) is a puzzle with 3 pegs, the object is to move a stack of disks from 1 peg to another while observing a set of rules. Here is a youtube video which explains it; and shows the rationale behind the below function. https://www.youtube.com/watch?v=YstLjLCGmgg

In a nut shell, the below function defines ToH. As you can tell it is recursive.

```
FUNCTION MoveTower(disk, source, dest, spare):
IF disk == 0, THEN:
    move disk from source to dest
ELSE:
    MoveTower(disk - 1, source, spare, dest)
    move disk from source to dest
    MoveTower(disk - 1, spare, dest, source)
END IF
```

Implement ToH in MIPS assembly using the constructions of AR implementation discussed in class (follow the conventions from class). Test your implementation with varying values of ***disk***. If it works with ***disk = 4*** then it should for any value.

Part B

Finding the roots to an equation using Newton's Method.

Objective of this program is to practice floating point calculations in MIPS. All calculations should be done as doubles. Note, that output will not be very eye appealing due to the format when doubles are printed. Don't worry about that.

Newton devised a method to find the/a root to an equation by applying the following formula until the error was sufficiently small.

$$x_{n+1} = x_n - (F(x_n)/F'(x_n))$$

Consider a continuous function $\mathbf{x^3-4x^2+1}$ where its derivative F' is $\mathbf{3x^2-8x}$. Thus, by calculating successive values of **x**, the value of $\mathbf{F(x_n)}$ will approach **0**, and hence be a root of **F**.

Write a MIPS program which will find $\mathbf{F(x_n)=0}$, to a tolerance of $\mathbf{10^{-6}}$. Thus, find $\mathbf{x_n}$. so that the change of $\mathbf{F(x_n)}$ is less than 10^{-6}.

To start you should write 2 leaf functions, one which calculates F and the other F'. Prompt the user to enter an initial value for **x**, hence $\mathbf{x_1}$. Then calculate each new **x** until **F(x)** is sufficiently close to **0**. Try **0.5** as an initial value.

Your program should output **x** and the value of **F(x)** for each root that you find. You should try other values for the initial value of **x** to test your program.

Submission

Submit full code solutions via Brightspace. Use proper code documentation. Marks are awarded for code readability.

end