COS 554 HW #2 Due Tuesday January 27, 2009

1. Prove by the Principle of Recursion that

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

Submit a listing of the program you wrote for the proof and enough output to suggest that the equation is correct.

2. Prove by the Principle of Recursion that

$$\sum_{i=1}^{n} \frac{i^2}{(2i-1)(2i+1)} = \frac{n(n+1)}{2(2n+1)}$$

Submit a listing of the program you wrote for the proof and enough output to suggest that the equation is correct.

3. Prove by the Principle of Recursion that the sum of the cubes of 3 consecutive integers is divisible by 9. Submit a listing of the program you wrote for the proof together with some output that illustrates how the program works.

4. Prove by the Principle of Recursion that for any integer $n \geq 0$,

$$11^{n+2} + 12^{2n+1}$$

is divisible by 133. Submit a listing of the program you wrote for the proof together with some output that illustrates how the program works. (Hint: A simple reduction into sums does not work for this problem. You should consider a scheme like the one used to solve a pair of linear equations in two unknowns. In particular, think about how you eliminate one variable from such a system of equations).

5. Use recursion and the definition of the Fibonacci numbers (Fib) to prove that

$$2^n \geq \text{Fib}(n) \geq 2^{n/2} \text{ for } n \geq 2$$

What happens for n = 0 and n = 1?

6. Let $f: \mathbb{Z} \rightarrow \mathbb{Z}$ be a function from the integers into the integers and $S \subseteq \mathbb{Z}$ a subset of the integers. S is said to be *preserved by f* if for each $d \in S$, $F(d) \in S$.

Now let f be given by

```
def f(n):
   if (n > 100):
     return (n - 10)
   else:
     return f(f(n+11))
```

Prove that the set of integers 91... is preserved by F. Use this fact to prove that recursion always halts when F is called.

7. (Pentagon Problem) The following shows it is not always easy to decide whether a recursive function will halt or not. To each vertex of a pentagon an integer is assigned is such a way that the sum of all five numbers is positive. Consider the following operation, OPR, on the pentagon: whenever a vertex is labeled with a negative integer, n <0, and if its neighbors are a and b, the triple of values a, n, b is replaced by a+n, -n, b+n respectively. For example, suppose the 5 vertices of the pentagon have the values -10 5 5 5 6, you can replace them with the set of values 10 -5 5 5 -4. Note that since there are two negative integers labeling the vertices of the pentagon there are two ways to apply the operation OPR.

   The question of interest is whether it is possible to come up with an initial set of 5 integers such that the operation OPR can be applied an infinite number of times? To help answer this question do the following:

a. Write a *recursive program* that calculates the largest number of times that OPR can be applied to a sequence of 5 numbers. Of course, for the sequences to which OPR can be applied infinitely many times, the program will not halt. Run the program on some quintuples and submit some sample runs and a listing.

b. Either prove that your program halts for all quintuples of integers whose sum is positive or give an example of a quintuple for which the program loops infinitely.

c. What is the answer to Part b) if the initial sum is allowed to be 0?

d. What is the answer to Part b) if the initial sum is allowed to be negative?

8. Consider the following recursive function.

```
def depth(n):
  if (n = = 1):
    return 1
  if (1 = = n % 2):
    return 1 + depth(3*n + 1)
  return 1 + depth(n/2)
```

   It is not known whether the above program will halt for every value of n, or if there are some values for which it will run infinitely. Use the above program as a model to write a program that verifies that for all integers between 1 and 100, the function depth halts. Submit a listing and a run of your program. Also find the value between 1 and 100 for which depth has its maximum value. Print out this value and the point at which it occurs.