

CIS-280
Assignment 1: Basic Concepts and Recursion: 10 pts. each
Due Tuesday, February 22, 2005

1. Replace *expression1*, *expression2*, *expression3*, and *expression4* with appropriate Scheme expressions to complete the definition of a procedure `EndOnes` that takes three numbers as arguments and returns the sum of the cubes of the largest and smallest numbers. **You must design the procedure as shown below, just filling in *expression1*, *expression2*, *expression3*, and *expression4*.** Note that *expression1* will be another `cond` expression, thus providing an example of nested conditionals.

```
(define EndOnes (lambda (x y z)
  (cond ((>= x y) expression1 )
        ((>= x z) expression2 )
        ((>= y z) expression3 )
        (else expression4))))
```

2. Define and test a recursive procedure (`ith-digit num i`) that takes as arguments a positive integer `num` and a non-negative integer `i`, and returns the digit of `num` that is in position `i` where the rightmost digit is viewed as in position 0. Thus

(`ith-digit 37562798 4`) returns 6

3. Define a recursive procedure (`SumAlt n`) that returns the sum of every other digit in its argument `n`, beginning with the digit in the units position. So for example, (`SumAlt 425638`) will return 16 (the sum of 8, 6, and 2).

4. The *funnynumbers* are 1,2,8,22,11,47,13,...

The first number is 1 and the second is 2; for $n > 2$, the n -th *funnynumber* can be computed as the square of n minus the absolute value of the difference between the two preceding *funnynumbers* if n is odd, and as the square of n plus the absolute value of the difference between the two preceding *funnynumbers* if n is even. For example, the 5th *funnynumber* is $5^2 - |(22 - 8)|$, and the 6th *funnynumber* is $6^2 + |(11 - 22)|$. Design and test a recursive Scheme procedure (`funnynumber n`) that takes a positive integer `n` as argument and returns the n -th *funnynumber*.

5. Define a recursive procedure (`SimilarDigits num1 num2`) that takes as arguments two positive integers `num1` and `num2`, and returns true if the corresponding digits of `num1` and `num2` are either both even or both odd, and returns false otherwise. For example,

(`SimilarDigits 7873564 5619522`) returns true
(`SimilarDigits 1133 9999`) returns true
(`SimilarDigits 2342 6425`) returns false

6. Define a recursive procedure (`SumDigits num1 num2`) that takes as argument two positive integers `num1` and `num2` and returns the sum of the product of the corresponding digits. For example,

(`SumDigits 345 221`) returns 19
(`SumDigits 3785 28`) returns 56
(`SumDigits 332 445233`) returns 21

Place your procedures in a single file, copy them into the submission web page, and submit them for grading.