Due Thursday October 7, 11:59 p.m. Don't wait until the last second!

Read each question carefully and write a short procedure to implement the function described. Don't be alarmed by all the math symbols; the actual functions are all quite simple and are completely described. Remember – unlike the text, use lowercase letters for all procedures.

Check that the language level in Dr.Scheme is set to R5RS. Then put all the procedures that you write into one Scheme file named hw1.scm. Be sure to name the procedures as requested in the question. This way it is easy to run standard test cases. A limited set of test cases is available in hw1-test-it.scm, provided along with this writeup. A sample execution of hw1-test-it is shown on the last page of this handout.

When you are done, follow the turnin link from the calendar page. You will need your UW NetId in order to do this. Follow the instructions and submit the file hw1.scm, **then complete the web survey that you are directed to.**

1. Write a procedure (volume-of-sphere r) that calculates the volume of a sphere given the radius r using the formula $V = \frac{4}{3}\pi r^3$. Use pi = 3.1415926535.

2. The greatest common denominator of two numbers *a* and *b* is easily found using the recursive Euclidean algorithm. The procedure takes two integer arguments and returns a single integer greater than or equal to 1. The operation of the procedure is as follows:

   if *b* is equal to 0, then return (abs *a*)
   otherwise calculate and return the greatest common denominator of *b* and the remainder from *a*/*b*

   Write a recursive procedure (euclid a b) that implements this algorithm. There is a predefined function (remainder x y) that you can use in your implementation. You can check your solution by comparing the results with those of the predefined function (gcd a b).

3. Write a procedure (maggy x y z) that takes three arguments representing the (x, y, z) coordinates of a vector and returns the length of the vector, where
   $\text{maggy}(x, y, z) = \sqrt{x^2 + y^2 + z^2}$

4. Some functions are defined recursively by giving the value of the function at a base point in the domain, and a rule for obtaining the next value from a previous value. Write a simple recursive procedure (funky s n) to compute f(s,n),

   where
   $$f(s,0) = s$$
   $$f(s,n+1) = \sqrt{f(s,n)}$$

   The domain is the non-negative integers and you can assume that s and n are both in the domain.

5. The binomial theorem states that

$$(x+y)^n = \sum_{j=0}^{n} C(n,j) x^{n-j} y^j$$
$$= \binom{n}{0} x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \cdots + \binom{n}{n-1} x y^{n-1} + \binom{n}{n} y^n$$

In other words, the coefficient of $x^{n-j}y^j$ is $C(n,j) = \binom{n}{j} = \frac{n!}{j!(n-j)!}$.

Copy one of the factorial functions from lecture and name it factorial.

Write a procedure (binom n j) that calculates the binomial coefficient C(n,j) for a given (n,j) combination. You can assume n,j ≥ 0 with j ≤ n.

6. Proper quadratic equations of the form $ax^2+bx+c=0$, a not equal 0, have zero, one, or two real roots as the discriminant $D=b^2-4ac$ is less than zero, equal 0, or greater than zero. Write a procedure (real-root-count a b c) that returns 0, 1, or 2 depending on the value of D. You can assume that a is not equal 0.

7. Creating the binary representation of a number can be done with repeated divisions and string concatenations. The algorithm is:

```
if a = 0 return "0"
if a = 1 return "1"
otherwise return expansion(a,"")
```

where expansion(a,str) is an iterative procedure that implements that following logic:

```
if a = 0
        return str
else
        return expansion(quotient(a/2),string-append("0" or "1",str))
```

The choice of “0” or “1” is based on the value of (remainder a 2). The procedure (quotient a b) returns the integer result of a/b.

Write a procedure (to-binary a) that returns a string representation of the binary value of a. For example, (to-binary 5) returns “101” and (to-binary 0) returns “0”.

```
volume-of-sphere
0 => 0.0 : 0.0
4.188790204666667 => 4.188790204666667 : 0.0
1 => 1.0000000439817005 : 4.3981700503792354e-008
3.1415926535 => 3.1415926535 : 0.0

euclid
2 => 2 : 0
10 => 10 : 0
10 => 10 : 0
7 => 7 : 0
1 => 1 : 0
7 => 7 : 0
7 => 7 : 0

maggy
1 => 1 : 0
1.4142135623730951 => 1.4142135623730951 : 0.0
1.7320508075688772 => 1.7320508075688772 : 0.0
5 => 5 : 0
5 => 5 : 0

funky
3 => 3 : 0
1.7320508075688772 => 1.7320508075688772 : 0.0
1.3160740129524924 => 1.3160740129524924 : 0.0
1.0 => 1.0 : 0.0

binom
1 => 1.0 : 0.0
2 => 2.0 : 0.0
10 => 10.0 : 0.0
1 => 1.0 : 0.0
70 => 70.0 : 0.0

real-root-count
0 => 0 : 0
1 => 1 : 0
2 => 2 : 0
2 => 2 : 0
2 => 2 : 0

to-binary
0 => 0 : #t
1111 => 1111 : #t
100000 => 100000 : #t
10101010 => 10101010 : #t
```