Exponents or Powers

Find a function that given a real number, a, and a positive integer, b, will return a^b .

```
double exp (double a; int b)
{ // Pre:  b ≥ 0

// Post:  result = a^b
} // exp
```

In mathematics:

$$x^0 = 1$$

$$x^k = x*x^{k-1} , k > 0$$

In Java:

```
double exp (double a, int b)
{ // Pre:  b ≥ 0
  if ( b == 0 )
        result = 1.0;
  else
        result = a*exp(a, b-1);
return result;
// Post:  result = a^b
} // exp
```

In effect, the program, exp, calculates x^k as: $x*x*\ldots*x$ (k occurrences of x).

This can be implemented by a ‘for loop’ in Java and the program calculates x^n in n steps.

```
double linear_exp(double a, int b)
   {// Pre: b ≥ 0
      double r = 1.0;

      for (int k = 1; k <= b ; k = k+1)
         r = r*a;
      return r;
      // Post: r = a^b
   } // linear_exp
```

Binary Exponent Function

In mathematics

$x^0 = 1$

$x^k = (x*x)^{k/2}$, if even(k)

$x^k = x*x^{k-1}$, if odd(k)

Note: $(x^a)^b = x^{a*b}$, therefore $(x^2)^{k/2} = x^{2*(k/2)} = x^k$.

also, $x^a * x^b = x^{a+b}$.

This leads to a more efficient program by calculating x^n in $\log_2(n)$ steps.

e.g. $\log_2(1024) = \log_2(2^{10})$. This program calculate 2^{1024} in about 10 steps.

In Java:

```
double f_exp(double a, int b)
{ // Pre: b >= 0
   double x = a;
   int k = b;
   double result;

   if ( k == 0 )
       result = 1;
   else if ( k%2 == 0 )
           result = f_exp(x*x, k/2);
        else
           result = x*f_exp(x, k-1);
   return result;
//Post: result = a^b
}
```

This recursive Java program can be rewritten using a ‘while loop’ as:

```
double fast_exp(double a, int b)
    {
        double x = a;
        int k = b;
        double r = 1.0;

        while (k != 0)
            if ( k%2 == 0 )
            {
                x = x*x;
                k = k/2;
            }
            else
            {
                r = r*x;
                k = k-1;
            }
        return r;
    //Post:  r = a$^b$
    } // fast_exp
```

Tracing the Program

```
double fast_exp(double a, int b)
   {
      double x = a;
      int k = b;
      double r = 1.0;

      while (k != 0)
         if ( k%2 == 0 )
         {
            x = x*x;
            k = k/2;
         }
         else
         {
            r = r*x;
            k = k-1;
         }
      return r;
   //Post:  r = a^b
   } // fast_exp
```

x	k	k==0	k%2 == 0	r
4	5			1
		FALSE		
			FALSE	
				4
	4			
		FALSE		
			TRUE	
16				
	2			
		FALSE		
			TRUE	
256				
	1			
		FALSE		
			FALSE	
				1024
	0			
		TRUE		