// Discrete Probability Library  
public class DProb  
{  
 public static double Permutations(long N, long X)  
 {  
 double perm = 1.0;  
   
 // P(N, X) = (N!)/((N-X)!)  
 for(long i=0; (N-i)> N-X; i++)  
 {  
 perm \*= (double)(N-i);  
 }  
 return perm;  
 }  
   
 //------------------------------------------------------  
   
 public static double Combinations(long N, long X)  
 {  
 double comb = 1.0;  
 long numLoops = 0;  
   
 if(X == 0 || X == N)  
 {  
 return comb;  
 }  
   
 // figure out which of the two versions can be reduced more // 12650  
 if(X > (N-X))  
 {  
 numLoops = N-X;  
 }  
 else   
 {  
 numLoops = N - (N-X);  
 }  
   
   
 for(long i=1; i<=numLoops; i++)  
 {  
 comb \*= (double) ( (N-i) + 1) / (i);  
 }  
   
 return comb;  
 }  
   
 //------------------------------------------------------  
  
 public static double HyperGeometric(long Np, long Xp, long N, long X)  
 {  
 double probX = 1.0;  
   
 probX \*= Combinations(Xp, X)\*Combinations(Np-Xp, N-X) / Combinations(Np, N);  
   
 return probX;  
 }  
   
 //------------------------------------------------------  
  
 public static double Binomial(double P, long N, long X)  
 {  
 double probX = 1.0;  
   
 probX \*= Combinations(N, X) \* Math.pow(P,X) \* Math.pow((1-P),(N-X));  
   
 return probX;  
 }  
   
 //------------------------------------------------------  
   
 public static double Poisson(double Xmean, long X)  
 {  
 double probX = 1.0;  
 double denominator = (double) X;  
  
 probX \*= Math.pow(Math.E, -Xmean) \* Math.pow(Xmean, X);  
   
 while(X > 1)  
 {  
 denominator \*= (X - 1);  
 X--;  
 }  
   
 probX /= denominator;  
   
 return probX;  
 }  
} // end class

class DprobTest{  
 public static void main(String args[])  
 {  
 System.out.println("1A: " + DProb.Permutations(33, 4));  
 System.out.println("1B: " + DProb.Combinations(33, 4) + "\n");  
   
 System.out.println("2A: " + DProb.HyperGeometric(50, 8, 9, 0));  
 System.out.println("2B: " + DProb.HyperGeometric(30, 8, 9, 0));  
 System.out.println("2C: " + (DProb.HyperGeometric(50, 8, 9, 0) \*   
 DProb.HyperGeometric(30, 8, 9, 0)) );  
 System.out.println("2D: " + DProb.HyperGeometric(80, 8, 13, 0) + "\n");  
  
 System.out.println("3A: " + DProb.HyperGeometric(71, 10, 5, 2));  
 System.out.println("3B: " + (DProb.HyperGeometric(71, 10, 5, 1) +   
 DProb.HyperGeometric(71, 10, 5, 0)) + "\n");  
   
 System.out.println("4A: " + DProb.Binomial(0.069, 30, 3));  
 System.out.println("4B: " + (DProb.Binomial(0.069, 30, 2) +   
 DProb.Binomial(0.069, 30, 1) +   
 DProb.Binomial(0.069, 30, 0)) + "\n");  
   
 System.out.println("5A Binomial: " + DProb.Binomial(0.026, 125, 4));  
 System.out.println("5B Binomial: " + (DProb.Binomial(0.026, 125, 3) +   
 DProb.Binomial(0.026, 125, 2) +   
 DProb.Binomial(0.026, 125, 1) +   
 DProb.Binomial(0.026, 125, 0)) + "\n");  
   
 System.out.println("5A Poisson: " + DProb.Poisson(0.026 \* 125, 4));  
 System.out.println("5B Poisson: " + (DProb.Poisson(0.026 \* 125, 3) +   
 DProb.Poisson(0.026 \* 125, 2) +   
 DProb.Poisson(0.026 \* 125, 1) +   
 DProb.Poisson(0.026 \* 125, 0)) + "\n");  
   
 }// end main  
}// end class

Results are on the next page down

||

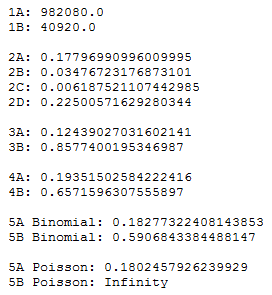
||

||

\_||\_

\ /

\/



Poisson was a good estimate for 5A but for some reason whenever I input 0 as an input for Poisson I would get Infinity.