**Part A**

// Discrete Probability Library  
  
public class DProb  
{  
  
 public static double Permutations(long N, long X)  
 {  
 double perm = 1.0;  
   
 for(long i = N - X + 1;i <= N; i++)  
 {  
 perm = perm \* (double)(i);   
 }  
  
 return perm;  
 }  
   
//end permutations  
  
  
 public static double Combinations(long N, long X)  
 {  
 double comb = 1.0;  
   
 long i;  
   
 if(X==0||X==N)  
 {  
 return comb;   
 }  
   
 if(N-X>X)  
 {  
 for(i=1;i<=X;i++)  
 {   
 comb = comb\*(double)(N-i+1)/(double)(i);  
   
 return comb;  
 }  
 }  
   
 else  
 {  
 for(i=1;i<=N-X;i++)   
 {  
 comb = comb\*(double)(N-i+1)/(double)(i);  
   
 return comb;  
 }  
 }  
   
 return comb;  
 }  
  
//end compbinations  
  
  
 public static double HyperGeometric(long Np, long Xp, long N, long X)  
 {  
 double probX = 0.0;  
   
 probX = ((Combinations(Xp, X) \* Combinations(Np - Xp, Np - X))/(Combinations(Np, N)));  
   
 return probX;  
 }  
  
//end HyperGeometr

ic  
  
  
 public static double Binomial(double P, long N, long X)  
 {  
 double probX = 0.0;  
  
 probX = (Combinations(N,X) \* Math.pow(P,X) \* Math.pow(1-P,N-X));   
  
 return probX;  
 }  
  
//end Binomial  
  
  
 public static double Poisson(double Xmean, long X)  
 {  
 double probX = 0.0;  
  
 probX = (Math.exp(-Xmean)\*Math.pow(Xmean,X)/(Permutations(X,X)));  
  
 return probX;  
 }  
  
//end Poisson  
  
  
public static void main(String args[])  
{  
System.out.println("Permutation: " + DProb.Permutations(10, 5));  
System.out.println("Combonation: " + DProb.Combinations(10,5));  
System.out.println("HyperGeometric: " + DProb.HyperGeometric(30,20,10,5));  
System.out.println("Binomial: " + DProb.Binomial(0.5,10,5));  
System.out.println("Poisson: " + DProb.Poisson(0.05,10));  
}  
  
public static double(  
  
  
} // end class

**Part B**

public static void main(String args[])  
{  
  
System.out.println("\nPart 1a "+DProb.Permutations(33, 4));  
  
System.out.println("\nPart 1b " + DProb.Combinations(33, 4));  
  
System.out.println("\nPart 2a " + DProb.HyperGeometric(50,9,8,0));  
  
System.out.println("\nPart 2b " + DProb.HyperGeometric(30,4,8,0));  
  
double hold = DProb.HyperGeometric(50,9,8,0)+DProb.HyperGeometric(30,4,4,0);  
  
System.out.println("\nPart 2c " + hold);  
  
System.out.println("\nPart 2d " + DProb.HyperGeometric(80,13,8,0));  
  
System.out.println("\nPart 3a " + DProb.HyperGeometric(71,5,10,2));  
  
hold = DProb.HyperGeometric(71,5,10,1) + DProb.HyperGeometric(71,5,10,0);  
  
System.out.println("\nPart 3b " + hold);  
  
System.out.println("\nPart 4a " + DProb.Binomial(0.069,30,3));  
  
hold = DProb.Binomial(0.069,30,2) + DProb.Binomial(0.069,30,1) + DProb.Binomial(0.069,30,0);  
  
System.out.println("\nPart 4b " + hold);  
  
System.out.println("\nPart 5a binomial " + DProb.Binomial(0.026,125,4));  
  
System.out.println("\nPart 5a poisson " + DProb.Poisson(3.25,4));  
  
hold = DProb.Binomial(0.026,125,3) + DProb.Binomial(0.026,125,2) + DProb.Binomial(0.026,125,1)+DProb.Binomial(0.026,125,0);  
  
System.out.println("\nPart 5b binomial " + hold);  
  
hold = DProb.Poisson(3.25,3) + DProb.Poisson(3.25,2) + DProb.Poisson(3.25,10)+ DProb.Poisson(3.25,0);  
  
System.out.println("\nPart 5b poisson " + hold);  
  
}

**Output**

Part 1a 982080.0  
  
Part 1b 40920.0  
  
Part 2a 1.862618303037381E-9  
  
Part 2b 1.7085474356838674E-7  
  
Part 2c 3.649155428040849E-5  
  
Part 2d 3.449758407640367E-11  
  
Part 3a 2.16573010170579E-11  
  
Part 3b 1.299438061023474E-11  
  
Part 4a 0.19351502584222413  
  
Part 4b 0.6571596307555897  
  
Part 5a binomial 0.18277322408143856  
  
Part 5a poisson 0.18024579262399287  
  
Part 5b binomial 0.5906843384488147  
  
Part 5b poisson 0.4667962652849057

**Part C**

1.

import java.util.Random;  
  
public class Binomial   
{  
  
private double P;   
private long N;   
  
public Binomial(double P, long N)   
{  
 this.N = N;  
 this.P = P;  
}  
  
public double getP()   
{  
 return P;  
}  
  
public void setP(double p)   
{  
 P = p;  
}  
  
public long getN()   
{  
 return N;  
}  
  
public void setN(long n)   
{  
 N = n;  
}  
  
long randomX()   
{  
 long y;  
   
 y = new Random().nextInt(100);  
   
 return y;  
}  
  
double mean()   
{  
 return N\*P;  
}  
  
double variance()   
{  
 return N\*P\*(1-P);  
}  
  
double probX(long x)   
{  
 return fact(N)/(fact(x)\*fact(N-x))\*(Math.pow((Math.PI),x))\*(Math.pow((1-Math.PI),(N-x)));  
  
  
}  
  
public static double fact(long n)   
{  
 double ans = 0.0;  
 for (int i = 1; i <= n; i++)  
 ans \*= i;  
 return ans;  
}  
  
public static void main(String...s)  
{  
Binomial binomial1 = new Binomial(0.36, 22);  
double N = binomial1.getN();  
double P = binomial1.getP();  
long X = binomial1.randomX();  
double mean = binomial1.mean();  
double variance = binomial1.variance();  
  
System.out.println("N = " + N);  
System.out.println("P = " + P);  
System.out.println("X = " + X);  
System.out.println("Mean = " + mean);  
System.out.println("Variance " + variance);  
}  
}

**Output**

N = 22.0  
P = 0.36  
X = 77  
Mean = 7.92  
Variance 5.0688

2.

import java.util.Random;  
  
public class Binomial   
{  
  
private double P;   
private long N;   
  
public Binomial(double P, long N)   
{  
 this.N = N;  
 this.P = P;  
}  
  
 public static double Permutations(long N, long X)  
 {  
 double perm = 1.0;  
   
 for(long i = N - X + 1;i <= N; i++)  
 {  
 perm = perm \* (double)(i);   
 }  
  
 return perm;  
 }  
  
public double getP()   
{  
 return P;  
}  
  
public void setP(double p)   
{  
 P = p;  
}  
  
public long getN()   
{  
 return N;  
}  
  
public void setN(long n)   
{  
 N = n;  
}  
  
long randomX()   
{  
 long y;  
   
 y = new Random().nextInt(100);  
   
 return y;  
}  
  
double mean()   
{  
 return N\*P;  
}  
  
double variance()   
{  
 return N\*P\*(1-P);  
}  
  
double probX(long x)   
{  
 double probX = 1.0;  
  
 probX = (Math.exp(x)\*Math.pow(x,x)/(Permutations(x,x)));  
  
 return probX;  
  
  
}  
  
public static double fact(long n)   
{  
 double ans = 0.0;  
 for (int i = 1; i <= n; i++)  
 ans \*= i;  
 return ans;  
}  
  
public static void main(String...s)  
{  
Binomial binomial2 = new Binomial(0.73, 6);  
System.out.println("XPr(X)");  
for(int i=0; i<=binomial2.getN(); i++) {  
long rand2 = binomial2.randomX();  
double prob = binomial2.probX(rand2);  
System.out.println(i+","+rand2+","+prob);}  
  
  
}  
}

**Output**

XPr(X)  
0,80,1.3678226440341753E68  
1,18,4.0351970357141475E14  
2,77,3.455770582145951E65  
3,61,4.915623902666711E51  
4,9,8651071.256694488  
5,6,26142.185818329235  
6,67,7.634719334185861E56  
