**11: Implementation of Chi Square Goodness of fit test for Poisson distribution.**

import java.util.\*;

class ChisquarePoisson

{

static double d()

{

return sc.nextDouble();

}

static int i()

{

return sc.nextInt();

}

static double po(double alpha , int x)

{

return (Math.pow(Math.E , -alpha)\*Math.pow(alpha , x))/fact(x);

}

static long fact(int f)

{

if(f == 0 || f == 1)

return 1;

long ans = 1;

for(int i=f; i>1; i--)

ans \*= i;

return ans;

}

static Scanner sc = new Scanner(System.in);

public static void main(String args[])

{

// System.out.println("Enter Number of random numbers");

// int n = i();

double random[] = new double[100];

// System.out.println("Enter seed value");

// double seed = d();

// random[0] = seed;

// System.out.println("Enter a");

// double a = d();

// System.out.println("Enter c");

// double c = d();

// System.out.println("Enter m");

// double m = d();

// for(int i=1; i<n; i++)

// {

// random[i] = ((a\*random[i-1]+c)%m)/m;

// random[i] = Math.floor(random[i]\*100)/100;

// System.out.println(random[i]);

// }

Random r = new Random();

for(int i=0; i<100; i++)

{

random[i] = (r.nextInt(100)\*1.0)/100;

System.out.println(random[i]);

}

int observed[] = new int[10];

double expected[] = new double[10];

for(int i=0; i<100; i++)

{

int tp = (int)Math.floor(random[i]\*10);

//System.out.println(tp);

observed[tp]++;

}

double alpha = 0, chi = 0;

for(int i=0; i<10; i++)

{

//System.out.println(observed[i]);

alpha += (i \* observed[i]);

}

alpha /= 100;

System.out.println(alpha);

System.out.println("\n\tno\tobserved\texpected");

for(int i=0; i<10; i++)

{

double ans = 100\*po(alpha , i);

expected[i] = ans;

System.out.println("\t"+i+"\t"+observed[i]+"\t"+ans);

//System.out.printf("%.3f\n", ans);

}

ArrayList<Double> ex = new ArrayList();

ArrayList<Integer> ob = new ArrayList();

boolean flag = false;

for(int i=0; i<10; i++)

{

double cur = expected[i];

int cur1 = observed[i];

if( cur < 5.0)

{

int j=i+1;

for(; j<10; j++)

{

flag = true;

cur += expected[j];

cur1 += observed[j];

if(cur >= 5.0)

{

ex.add(cur);

ob.add(cur1);

break;

}

}

i = j;

if(!flag)

{

ex.add(cur);

ob.add(cur1);

}

}

else{

ex.add(cur);

ob.add(cur1);

}

}

System.out.println("modified tables");

for(int i=0; i<ex.size(); i++)

{

System.out.println("\t"+i+"\t"+ob.get(i)+"\t"+ex.get(i));

//System.out.printf("%.3f\n", );

}

int dof = ex.size()-2;

System.out.println("dof = "+dof);

for(int i=0; i<ex.size(); i++)

{

//System.out.println("\t"+i+"\t"+ob.get(i)+"\t"+ex.get(i));

//System.out.printf("%.3f\n", );

chi += Math.pow((ex.get(i)-ob.get(i)) , 2);

}

chi /= 100;

System.out.println("chi calculated = "+chi);

if(chi < 10)

System.out.println("Accepted");

else

System.out.println(" Not Accepted(Not poisson distributed)! try another distribution");

}

}