Report for Question No 4

1. The code has been written in Java.
2. All challenges with HW=i(i=3,4,5,6) were generated.
   Example of HW=3 generation:

   ```
   double sum=0;
   int counter=0;
   for(int a=0;a<62;a++)
    {
     for(int b=a+1;b<63;b++)
     {
      for(int c=b+1;c<64;c++)
      {
        sum+= HW3(a,b,c,freq);
        counter++;
       }
     }
   }
   ```

3. Sample output for 1 run of code:(Frequencies generated and printed are not shown here)

   ```
   Accuracy for n=3 is 0.6678907450076805
   Accuracy for n=4 is 0.5065677645992295
   Accuracy for n=5 is 0.5678095857151251
   Accuracy for n=6 is 0.5019974559838903
   ```

4.

HW	Theoretical Bias	Obtained Bias
3	66.67	66.78
4	50.00	50.65
5	60.00	56.78
6	53.84	50.19

5. For HW=3 , we found out some problem with the algorithm. The obtained accuracy was exactly equal to 100-predicted accuracy.
   We figured out that the prediction will have to be inverted.
   Example: Let q=80.9 for HW=3 (1110000...)
   Response should be 0.
   Q1+Q2+Q3=80.9   (Restriction)
   Decimal Value Sum of all 3 has to lie in 0.9 and 2.9 and not 0 and 3.
   Hence the prediction as per algorithm gave us 33% accuracy which was due to the fact that the parity of the integral parts of Q1, Q2,Q3 was odd with 66.67% chances (i.e the integral part of sum of Q1, Q2,Q3 was 79 66.67% of the time.
   So we just simply inverted our prediction.
   Of course the reason behind this is that the value of the fractional part and the integral part is somehow also controlled by the inherent distribution of the frequencies(Gaussian distribution here)