

```

package javaapplication9;
import java.io.*;
import java.util.*;

public class JavaApplication9 {

    public static void main(String[] args)throws IOException
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        //Generating Frequencies
        double freq[]=new double[64];
        Random r=new Random();
        for (int i=0;i<64;i++)
        {
            freq[i]=20000-r.nextGaussian();
            System.out.println(freq[i]);
        }
        //Hamming wt 3
        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++;
                }
            }
        }
        System.out.println("Accuracy for n=3 is "+sum/counter);
        //Hamming wt 4
        double sum1=0;
        int counter1=0;
        for(int a=0;a<61;a++)
        {
            for(int b=a+1;b<62;b++)
            {
                for(int c=b+1;c<63;c++)
                {
                    for(int d=c+1;d<64;d++)
                    {
                        sum1+= HW4(a,b,c,d,freq);
                        counter1++;
                    }
                }
            }
        }
    }
}

```

```

}
double accuracy=sum1/counter1;
accuracy=accuracy<0.5?1-accuracy:accuracy;
System.out.println("Accuracy for n=4 is "+accuracy);
//Hamming wt 5
double sum2=0;
int counter2=0;
for(int a=0;a<60;a++)
{
    for(int b=a+1;b<61;b++)
    {
        for(int c=b+1;c<62;c++)
        {
            for(int d=c+1;d<63;d++)
            {
                for(int e=d+1;e<64;e++)
                {
                    sum2+= HW5(a,b,c,d,e,freq);
                    counter2++;
                }
            }
        }
    }
}
System.out.println("Accuracy for n=5 is "+sum2/counter2);
//Hamming wt 5
double sum3=0;
int counter3=0;
for(int a=0;a<59;a++)
{
    for(int b=a+1;b<60;b++)
    {
        for(int c=b+1;c<61;c++)
        {
            for(int d=c+1;d<62;d++)
            {
                for(int e=d+1;e<63;e++)
                {
                    for(int f=e+1;f<64;f++)
                    {
                        sum3+= HW6(a,b,c,d,e,f,freq);
                        counter3++;
                    }
                }
            }
        }
    }
}
System.out.println("Accuracy for n=6 is "+sum3/counter3);

```

```
}
```

```
static int HW3(int a,int b,int c,double[] freq)
```

```
{
```

```
    double q=findQ(a,b,freq)+findQ(b,c,freq)+findQ(c,a,freq);
```

```
    int k=(int)q;
```

```
    int sign=k%2;
```

```
    //now calculate predicted
```

```
        int sign1= (findQ1(a,b,freq)+findQ1(a,c,freq)+findQ1(c,b,freq))%2;
```

```
    int decision=0;
```

```
    if(sign!=sign1)
```

```
        decision=1;
```

```
    return decision;
```

```
}
```

```
static int HW3Correct(int a,int b,int c,double[] freq)
```

```
{
```

```
    double q=findQ(a,b,freq)+findQ(b,c,freq)+findQ(c,a,freq);
```

```
    int k=(int)q;
```

```
    int sign=k%2;
```

```
    return sign;
```

```
}
```

```
static int HW4(int a,int b,int c,int d,double[] freq)
```

```
{
```

```
    double q=findQ(a,b,freq)+findQ(a,c,freq)+findQ(a,d,freq)+findQ(b,c,freq)+findQ(b,d,freq)
```

```
+findQ(c,d,freq);
```

```
    int k=(int)q;
```

```
    int sign=k%2;
```

```
    //now calculate predicted
```

```
        int sign1= (findQ1(a,b,freq)+findQ1(a,c,freq)+findQ1(a,d,freq)+findQ1(b,c,freq)
```

```
+findQ1(b,d,freq)+findQ1(c,d,freq))%2;
```

```
    int decision=0;
```

```
    if(sign!=sign1)
```

```
        decision=1;
```

```
    return decision;
```

```
}
```

```
static int HW4Correct(int a,int b,int c,int d,double[] freq)
```

```
{
```

```
    double q=findQ(a,b,freq)+findQ(a,c,freq)+findQ(a,d,freq)+findQ(b,c,freq)+findQ(b,d,freq)
```

```
+findQ(c,d,freq);
```

```
    int k=(int)q;
```

```
    int sign=k%2;
```

```
    return sign;
```

```
}
```

```
static int HW5(int a,int b,int c,int d,int e,double[] freq)
```

```
{
```

```
    double q=findQ(a,b,freq)+findQ(a,c,freq)+findQ(a,d,freq)+findQ(a,e,freq)
```

```
        +findQ(b,c,freq)+findQ(b,d,freq)+findQ(b,e,freq)+
```

```
        findQ(c,d,freq)+findQ(c,e,freq)+findQ(d,e,freq);
```

```

    int k=(int)q;
    int sign=k%2;
    //now calculate predicted
    int sign1= (findQ1(a,b,freq)+findQ1(a,c,freq)+findQ1(a,d,freq)+findQ1(a,e,freq)
+HW4Correct(b,c,d,e,freq))%2;
    int decision=0;
    if(sign==sign1)
        decision=1;
    return decision;
}
static int HW6(int a,int b,int c,int d,int e,int f,double[] freq)
{
    double q=findQ(a,b,freq)+findQ(a,c,freq)+findQ(a,d,freq)+findQ(a,e,freq)+findQ(a,f,freq)
        +findQ(b,c,freq)+findQ(b,d,freq)+findQ(b,e,freq)+findQ(b,f,freq)+
        findQ(c,d,freq)+findQ(c,e,freq)+findQ(c,f,freq)+findQ(d,e,freq)+findQ(d,f,freq)
+findQ(f,e,freq);
    int k=(int)q;
    int sign=k%2;
    //now calculate predicted
    int sign1= (findQ1(a,b,freq)+findQ1(a,c,freq)+findQ1(a,d,freq)+findQ1(a,e,freq)
+findQ1(a,f,freq)
        +findQ1(b,c,freq)+findQ1(b,d,freq)+findQ1(b,e,freq)+findQ1(b,f,freq)+
        findQ1(c,d,freq)+findQ1(c,e,freq)+findQ1(c,f,freq)+
        HW3Correct(d,e,f,freq))%2;
    int decision=0;
    if(sign==sign1)
        decision=1;
    return decision;
}
static double findQ(int a,int b,double[] freq)
{
    return Math.abs((b-a)*Math.pow(Math.abs(freq[a]-freq[b]),0.5));
}
static int findQ1(int a,int b,double[] freq)
{
    double q= Math.abs((b-a)*Math.pow(Math.abs(freq[a]-freq[b]),0.5));
    int k=(int)q;
    int sign=k%2;
    return sign;
}
}

```