package Codes;

import java.io.BufferedInputStream;

import java.io.FileInputStream;

import java.io.FilterInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.util.Arrays;

public class gCorr

{

public double[] finalarr;

public gCorr(String name, int ws, int ss, int gl)

{

InputStream is = null;

FilterInputStream fis = null;

int p=0,k=0,l=0,j=0,count1=0,count2=0,a=0,s=0,len1=0,Ck=0,len2=0;

//int n=10;

//int gl=3053206;

int n=(int)(((gl-ws)/ss)+1);

char c;

double calc,Cg;

byte[] buffer = new byte[ws];

double arr[]=new double[n];

//System.out.println(name);

//System.out.println(ws);

//System.out.println(ss);

//System.out.println(gl);

try

{

// create input streams

is = new FileInputStream(name);

fis = new BufferedInputStream(is);

for(p=0;p<n;p++){

len1 = fis.read(buffer, 0,ss);

fis.mark(ws);

len2=fis.read(buffer,ss,ws-ss);

for(j=0;j<ws;j++)

{

//System.out.println("the array= " + (char)buffer[j]);

if((char)buffer[j]=='G')

buffer[j]=1;

else

buffer[j]=-1;

}

Cg=0;

for(k=1;k<ws;k++)

{

Ck=0;

for (l=1;l<ws-k+1;l++)

{

Ck=Ck+(buffer[l-1])\*(buffer[l+k-1]);

}

// System.out.println("Ck="+(Ck));

Cg=Cg+(double)((Math.abs(Ck+0.0))/(ws-k));

//System.out.println("Cg="+(Cg));

}

//System.out.println("value="+((Cg+0.0)/(ws-1)));

arr[s]= ((Cg+0.0)/(ws-1));

s++;

fis.reset();

// System.out.println("we have performed the reset!");

}

}

catch(IOException e)

{

// if any I/O error occurs

e.printStackTrace();

}

finally

{

// releases any system resources associated with the stream

if(is!=null)

try {

is.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

if(fis!=null)

try {

fis.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

//System.out.println(Arrays.toString(arr));

finalarr=arr.clone();

}

public double[] gcorrfunction()

{

//System.out.println(Arrays.toString(finalarr));

return finalarr;

}

public static void main(String[] args) throws Exception

{

}

}