1 Problem 1

By definition the golden mean $\frac{a}{b}$ satisfies $\frac{a}{b} = \frac{a+b}{a}$. Show that the golden mean has the value $\frac{1+\sqrt{5}}{2}$. Then modify the Fibonacci program to compute and write out the ratio of each term to its predecessor. (Don't forget that you want this ratio to be a double variable.) This $g(n)$ approaches a limit for large n. Show analytically that this limit is the golden mean defined above (hint: start from the definition: $f(n) = f(n-1) + f(n-2)$.)

The golden ratio is defined as $\phi = \frac{a}{b} = \frac{a+b}{a}$. This ratio can be solved for algebraically:

$$\begin{aligned}
a^2 &= ab + b^2 \\
a^2 - ab - b^2 &= 0 \\
a &= \frac{b \pm \sqrt{b^2 + 4b^2}}{2} \\
\phi = \frac{a}{b} &= \frac{1+\sqrt{5}}{2}.
\end{aligned}$$

In the last step, the plus sign was taken in order to keep with the definition that $\phi > 0$.

The Fibonacci sequence is given by

$$f(n) = \begin{cases} 1 & n = 0, 1 \\ f(n-1) + f(n-2) & n > 1 \end{cases}$$

Consider the ratio

$$r(n) = \frac{f(n)}{f(n-1)}.$$

Assume that $\lim_{n\to\infty} r(n) = c$. Claim that $c = \phi$.

Proof. (*Informal*) As n is arbitrarily large, additionally assume

$$\frac{f(n+1)}{f(n)} = \frac{f(n)}{f(n-1)} = c.$$

Then,

$$\begin{aligned}
c &= \frac{f(n+1)}{f(n)} \\
&= \frac{f(n-1) + f(n)}{f(n)} \\
&= \frac{1}{c} + 1 \\
c^2 - c - 1 &= 0 \\
c &= \frac{1+\sqrt{5}}{2} \\
c &= \phi
\end{aligned}$$

□

2 Problem 2

> Define a variable of type double to be the theoretical golden mean from the previous problem (to take the square root, use Math.sqrt) . Modify Fibonacci once again to display the difference between j/i and the golden mean. Let us call this difference for the n'th Fibonacci number q(n). We know from the previous problem that q(n) is supposed to approach zero as n gets large. But how does it approach zero? Plot q(n) vs n for n ranging from 3 to 20. You don't have to use the computer to make the plot. By plotting log q(n) vs n, test the hypothesis that q diminishes exponentially with n i. e. q(n) = (constant) exp(-n/N), where N is some "decay constant". From your graph, obtain an approximate value of N if this makes sense. You can use Java to take natural logs by using the function Math.log(number). You should also save the results to a file. This file should be handed in with the other files of you solution. If you have access to and are familiar with a program like Excel, MatLab, gnuPlot, etc., you can use those to plot your output for you.

For this problem, I modified the source code from Problem 1 to calculate $|\phi - r(n)|$ (modified source code included below and in folder P2). This difference should approach zero as n gets large, as indicated by the previous proof. The results of this calculation are plotted in Figure 1. The data is provided in the table beside the plot, and is accessible in the file `P2/output.txt`.

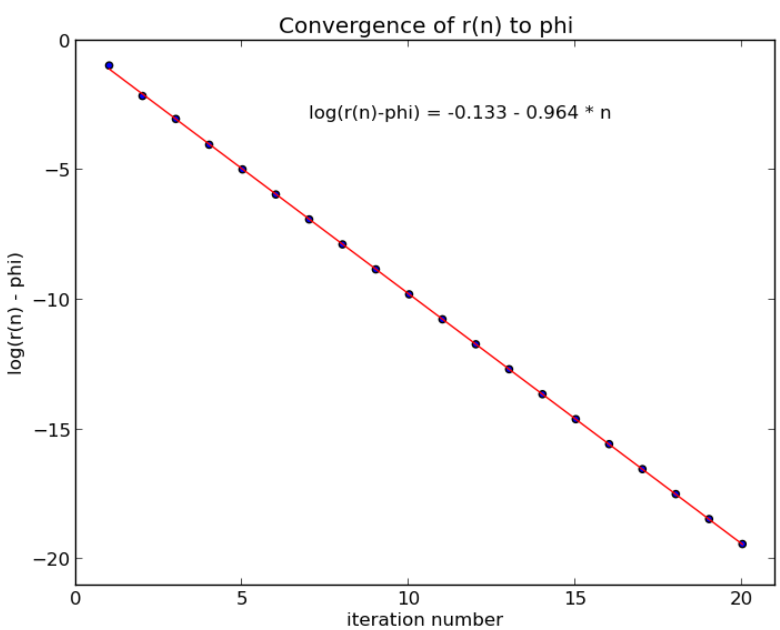

Figure 1: Convergence Plot

n	$q = \|r(n) - \phi\|$
1	3.82e-01
2	1.18e-01
3	4.86e-02
4	1.80e-02
5	6.97e-03
6	2.65e-03
7	1.01e-03
8	3.87e-04
9	1.48e-04
10	5.65e-05
11	2.16e-05
12	8.24e-06
13	3.15e-06
14	1.20e-06
15	4.59e-07
16	1.75e-07
17	6.70e-08
18	2.56e-08
19	9.77e-09
20	3.73e-09

As expected, $r(n)$ converges to ϕ exponentially with a fitted decay constant of $\delta = -0.964$. Thus,

$$r(n) = -.133e^{-.964n}.$$

3 Drawing

Figure 2 a screenshot of the drawing I made using the P251 helper class. The .java file I used to produce this is listed below Fibo.java.

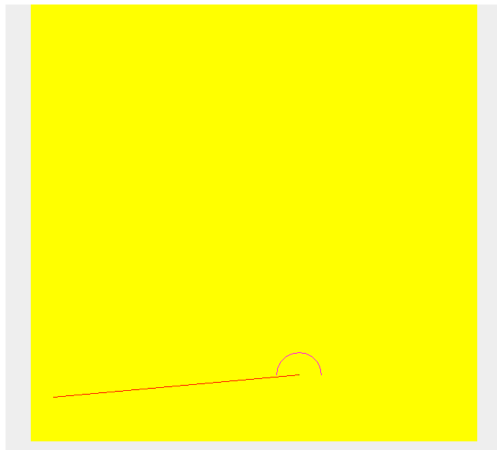

Figure 2: Screenshot of drawing

4 Appendix: Source Code

P2/Fibo.java

```
// Fibo.java
// Fibonacci java class 2
// write out fibonacci sequence
// ratio of terms, and convergence
// Athanasios Athanassiadis
// Complexity, Project 1, F2013
// 10/3/2012

import java.applet.Applet;
public class Fibo extends Applet
{

    int i,j,k;
    int n, nsteps, ypos;
    double r, q;
    double g = (1 + Math.sqrt(5)) / 2; //

    public void start()
    {
    nsteps = 30;

    i=1;
    j=1;
        System.out.println(" 0,1,1 " + (Math.abs(g-1)));

    for (n=1; n<=nsteps; n++)
    {
        k = i+j;
        r = (double) k / (double) j;
        q = Math.abs(g - r);
        System.out.println(" " + n + "," + k + "," + r + "," + q);
        i = j;
        j = k;
    }
    }

}
```

Draw/Drawing.java

```java
// Drawing.java
// first class to test drawing
import javax.swing.*;
import java.awt.Color;
import P251.*;

public class Drawing extends P251Applet
{

    private drawPanel dp;

    public void fillPanels()
    {
    dp = new drawPanel(600,600);
    dp.setBackgroundColor(Color.yellow);
    addPanel(dp);
    }

    public void compute()
    {
    dp.setDrawBounds(0f,0f, 100f, 100f);
    dp.addLine(5,10, 60,15, Color.red);
    dp.addArc(60-5,15+5, 10,10, 0,180, Color.magenta);
    dp.addLine(60,15, 60,15, Color.black);
    dp.repaint();
    }

}
```