Scripting Languages (Örgün + İÖ) Project # 2
(Version 1.2 - Updated 12/19 15:22)

Istanbul University - Computer Engineering Department

Email: Jan 1st 23:59, Hardcopy: Jan 5th, 17:00

Fibonacci Sequence Extractor

Fibonacci numbers or Fibonacci sequence is an integer sequence in which every number in the sequence is the sum or two preceding numbers. The sequence starts with `1, 1, 2, 3, 5, 8, 13, ...` and goes on like this. The closed form solution for the nth number in the sequence can be found using:

$$F_n = \frac{\varphi^n - (-\varphi)^{-n}}{\sqrt{5}}$$

where φ is called the "Golden Ratio" and its value is:

$$\varphi = \frac{1+\sqrt{5}}{2} \approx 1.6180339887\cdots$$

The position (index), n, of a given Fibonacci number, F_n, can be found using:

$$n = \left\lfloor \log_\varphi \left(F_n \cdot \sqrt{5} + \frac{1}{2} \right) \right\rfloor$$

$\lfloor x \rfloor$ is called the *floor* function, which returns the largest integer smaller that x. In Perl, `log` function will calculate the natural logarithm of a given argument. You can find the logarithm with base φ using

$$\log_\varphi(x) = \frac{\log(x)}{\log(\varphi)}$$

In this project you will create a script thet will take a text file consists of a long string and extracts a Fibonacci sequence from it, if one exists. Following is a description of the project. You must follow these rules to receive credit.

1. In your code, the strictures and warnings must be enabled. That is, put the following line in your preamble:

```
use strict;
use warnings;
```

2. The input file must fed as an invocation argument to string as the following:

```
% perl youscript.pl inputfile
```

If there is no argument, your program should then ask the user for the filename via STDIN only once. If no filename entered, the programs should warn the user and exits.

3. The input file has long string consists of numbers. A typical file has a string like the following (no line breaks).

```
0800428187847480471105608663939913025719016328560609473343200799833143548703086098289561305589144233377610987159725844181676543892778209312086712168146023724292592922103188415500572448653755337456341240720872678664590277249744322283465117806617824729341113006896854959612758363212893939265161695186881779976527733396308190979549954957632065089241725214523544495705372122949408856507917528315186617767606987367764988492151342299117361845845397452270900758972826200045221374874657423616656478482040441549251551264975028282045904494131755862615740367428721441575100461511079800356928914422408684041742573548270704010310596086222216967667001376721017038362746445578701970543229943636119608094023449969669109853908927149180320603233604443356143653514667126016685431430916203909103955972696809897897997565087398300408810056363350363531540704800487637226593946788864```

4. You will search for a Fibonacci sequence which may start at any point in the string. The Fibonacci sequence in above example is (shown in red above)

```
55 89 144 233 377 610 987 1597 2584 4181 6765
```

5. There may be more than one sequence in the file. You must find them all.

6. In this project, the first number in any sequence will not be more than 6 digits. That should help.

7. You will create two different methods to find the sequence. In the first version you will use the above formula to aid you in your search. You should use `int` function to prevent rounding errors.

8. For the second method, you will use recursive definiton of the Fibonacci sequence. That is:

$$F_n = F_{n-1} + F_{n-2}$$
$$F_1 = 1$$
$$F_2 = 1$$

To find whether a number belongs to the Fibonacci sequence, you may use a property of the Fibonacci numbers: A number, F, is a Fibonacci number if and only if one or both $5F^2 + 4$ or $5F^2 - 4$ is a perfect square. You can use this property to find check whether the first number is a Fibonacci number, but for the rest of the sequence you must use the recursive definition.

9. Do not use lookup tables (in which you store the whole sequence of Fibonacci numbers up to a point and match against the contents of the table.) You should use regular expressions and loops if you can.

10. Ignore the sequences that has less than 3 numbers. (For example is you find `58`, even though `5` and `8` are Fibonacci numbers, you will ignore them unless it is followed by `13`).

11. You will print the results on the screen.

12. You may use `floor` function from `POSIX` package (Usage: Put `use POSIX 'floor';` in the preamble). YOU MAY NOT USE ANY ADDITIONAL PERL PACKAGES. Your project must make use of only built-in Perl operations.
```

13. **You must add your name, number, project name and the date to the beginning of the script in comment lines**. Your grades will be based on comments (in English), clarity, efficiency, whether or not it can run, error catching etc. YOU MUST NAME your script file as the following: `schoolnumber_sl2017finalproject.pl`. (EXACT CAPITALIZATION!). (For example: `130600001_sl2017finalproject.pl`)

14. E-mail the file to before the due date to **dagtekin@istanbul.edu.tr** which MUST have a subject line: `sl2017finalproject`. Message body will be disregarded. CC to yourself and keep a copy of the file for your records. You can add additional stuff to the subject line to send updated code, ask questions etc. (For ex: `sl2017finalproject - updated script`)

15. Your submissions will be checked against each other for similarities using Stanford University's MOSS (Measure of Software Similarity) service. Do not cheat!

16. YOU MUST RETURN A PRINTED AND SIGNED COPY BY Januay 5^{th}, 17:00! You will not be given a grade if I don't receiev this.