

Program na01: Ratios of Fibonacci-like sequences

Overview: Write a program that computes Fibonacci-like sequences of numbers along with successive ratios.

Introduction: Medieval mathematician and businessman Fibonacci (Leonardo Pisano) posed the following problem in his treatise Liber Abaci (pub. 1202):

How many pairs of rabbits will be produced in a year, beginning with a single pair, if in every month each pair bears a new pair which becomes productive from the second month on?

It is easy to see that 1 pair will be produced the first month, and 1 pair also in the second month (since the new pair produced in the first month is not yet mature), and in the third month 2 pairs will be produced, one by the original pair and one by the pair which was produced in the first month. In the fourth month 3 pairs will be produced, and in the fifth month 5 pairs. After this things expand rapidly, and we get the following sequence of numbers:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...

This is an example of a recursive sequence, obeying the simple rule that to calculate the next term one simply sums the preceding two:

$F(1) = 1; \quad F(2) = 1; \quad F(n) = F(n-1) + F(n-2) \text{ for } n \geq 3.$

Thus 1 and 1 are 2, 1 and 2 are 3, 2 and 3 are 5, and so on.

Source: <http://www.mathacademy.com/pr/prime/articles/fibonac/index.asp>

If you look closely at the ratios of successive Fibonacci numbers, you are in for a big surprise! It turns out that the sequence of ratios $F(n)/F(n-1)$ converges to the number

$$\gamma = \frac{1 + \sqrt{5}}{2} = 1.6180339885...$$

The Greeks called this number the "golden ratio", and it goes by that name today.

The object of this program is to determine what effect, if any, the first two numbers have on the limit of the ratios of successive terms in the sequence.

Input: Use the standard input (i.e. the terminal). Each line of input will contain contains a separate case. Each case consists of three positive integers ≤ 100 with one or more spaces in between. The numbers are $F(1)$ and $F(2)$ and n , respectively. A blank line follows the last case and terminates input.

Example Input:

```
1 1 5
72 12 3
<blank>
```

Output: For each case, print a title and a header followed by the sequence of n Fibonacci-like numbers starting with $F(1)$ and $F(2)$, along with successive ratios. There should be a blank line at the end of each case. The title is

The first $<n>$ numbers and their successive ratios.

The header is

n $F(n)$ $F(n)/F(n-1)$

The first entry in the table has no ratio, since $F(0)$ is undefined.

In case $n > 1$, the k th entry of the sequence ($2 \leq k \leq n$) should contain k , $F(k)$ and the ratio $F(k)/F(k-1)$. Make sure that each column looks nice even when $n = 100$.

Example Output:

The first 5 numbers and their successive ratios.

n	$F(n)$	$F(n)/F(n-1)$
1	1	
2	1	1.0
3	2	2.0
4	3	1.5
5	5	1.66666667

The first 3 numbers and their successive ratios.

n	$F(n)$	$F(n)/F(n-1)$
1	72	
2	12	1.66666667
3	84	7.0

Program Specifications: Name the program file **na01<lastname>.py**. Input and output are to the standard input and output files. Your program must have a subroutine **fib(a,b,n)** that takes three integers **a**, **b**, **n** and outputs the appropriate Fibonacci-like sequence as a list. A second subroutine **printSequence(flist)** should print the list. Your main program should input each case, and should call the other routines. If you decide to do the extra credit part, only the printing routine should be changed.

Writeup: Use your program to answer the question: What is the effect of various inputs on successive ratios of Fibonacci-like sequences? Give relevant examples to support your conclusion. Your writeup should be in the file **na01<lastame>.txt**. It should have your name at the top and should be solely your work.

Extra for experts: na01X. Generate the same title and sequence, but make two columns in your table. Be sure that $\text{length}(\text{column } 2) \leq \text{length}(\text{column } 1) \leq \text{length}(\text{column } 2) + 1$. For example, the input 1 1 5 would result in the following output.

The first 5 numbers and their successive ratios.

n	$F(n)$	$F(n)/F(n-1)$	n	$F(n)$	$F(n)/F(n-1)$
1	1		4	3	1.5
2	1	1.0	5	5	1.66666667
3	2	2.0			

Checklist: Submit the following:

Program File: **na01.py** OR **na01X.py**

Writeup: **na01.txt**