

Zlatý řez a Fibonacciho posloupnost

Jirka je vášnivým fotografem, ale pro určení zlatého řezu a tedy i správné kompozice fotky potřebuje znát zlatý řez a Fibonacciho posloupnost. Pomoz mu napsat program, co ho vypočítá za něj.

Fibonacciho posloupnost

N-té Fibonacciho číslo je součet předchozích dvou Fibonacci čísel ($Fib(n-1) + Fib(n-2)$).

Začátek Fibonacciho sekvence:

- 0 (definováno)
- 1 (definováno)
- 1 (0 + 1)
- 2 (1 + 1)
- 3 (1 + 2)
- 5 (2 + 3)

Výpočet n-tého Fibonacciho čísla: $Fib(n) = \begin{cases} 0 & , n = 0; \\ 1 & , n = 1; \\ Fib(n-1) + Fib(n-2) & \text{jinak.} \end{cases}$

Nápověda: Rekurentní (rekurzivní) způsob výpočtu Fibonacciho čísla (ten nejjednodušší, vyzobrazený výše) je pro velká n velmi časově a výpočetně náročný. Pokuste se vymyslet lepší algoritmus nebo vylepšit a zrychlit ten výše zmíněný, jinak neprojdete složitými testy.

Zlatý řez

$$\text{Výpočet zlatého řezu: } GoldenRatio = \frac{Fib(n)}{Fib(n-1)} ; n \geq 1$$

Formát vstupu a výstupu

Na jednotlivých řádcích vstupu budete dostávat reálná čísla k a program na každý z nich váš program bude odpovídat dvěma hodnotami na jednom řádku oddělenými mezerou - k -tým Fibonacciho číslem a jeho zlatým řezem (k a $k-1$).

Při neplatném vstupu vypište `Invalid input`. Jako neplatný vstup se berou:

- čísla menší než nula (záporné indexy)
- desetinná čísla (desetinné indexy)

Pokud ze zadaného čísla nebude matematicky možné vypočítat zlatý řez, vypište místo jeho hodnoty `No golden ratio`. Pokud bude zlatý řez možné vypočítat vypište ho zaokrouhlený na 2 desetinná místa. Pokud vyjde "hezky" vypište např 5.00

Pokud budete úlohu řešit v programovacím jazyce C/C++/C# nebo v jakýkoliv jiném staticky typovaném jazyce, nezapomeňte pro výsledky použít 64-bitová desetinná čísla (`double` v C/C++/C#/Javě atd. V Pythonu, JavaScriptu a jim podobným jazycích toto řešit nemusíte.)

Příklady

Příklad 1

Vstup

```
3
5
13
```

Výstup

```
2 2.0
5 1.67
233 1.62
```

Příklad 2

Vstup

```
1
15
0
23
```

Výstup

```
1 No golden ratio
610 1.62
0 No golden ratio
28657 1.62
```

Příklad 3

Vstup

```
12.2
-6.8
22
-12
```

Výstup

```
Invalid input
Invalid input
17711 1.62
Invalid input
```

V této úloze je výrazně těžší získat všechny body - 3/4 bodů jsou jednodušší než bod čtvrtý

Veškeré výukové materiály se nachází na <https://github.com/delta-cs/seminar>.

V případě, že vám řešení nebude uznáno a nebudete vědět proč, navštivte nejdříve [stránku s technickými pravidly](#) a ujistěte se, že program všechna splňuje.

Veškeré výukové materiály se nachází na <https://github.com/delta-cs/seminar>.

V případě, že vám řešení nebude uznáno a nebudete vědět proč, navštivte nejdříve [stránku s technickými pravidly](#) a ujistěte se, že program všechna splňuje.