1 Zadanie 1

Zrealizuj ciąg fibonacciego zadany wzorem:

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

w podane sposoby:

1. Rekurencyjnie
2. Iteracyjnie (za pomocą pętli)
3. *Za pomocą 3 zmiennych

2 Zadanie 2*

Dany jest wzór:

$$f(K) = \begin{cases} 0 & \text{dla } K = 0 \\ f(K-1) + K & \text{dla } K > 1 \end{cases}$$

Program przyjmuje parametr N i zwraca liczbę x taką, że

$$f(x) \leqslant N$$

$$f(x+1) > N$$

.
np. dla N = 17 program powinien zwrócić 5, ponieważ:
f(0) = 0
f(1) = 1
f(2) = 3
f(3) = 6
f(4) = 10
f(5) = 15
f(6) = 21

3 Wytłumaczenie zadania 2

Najprostrzą metodą jest zaimplementowanie funkcji rekurencyjnie dokładnie tak jak jest zadana, ale czy to jest jedyny sposób i przede wszystkim czy to jest najlepszy sposób? Otóż okazuje się, że NIE.
Aby można było to zadanie rozwiązać optymalnie potrzebna jest teoria matematyczna, ale nie ma czego się bać - nie jest to trudna teoria.
Na podstawie tego co jest zapisane powyżej w rozwinięciu funkcji zobaczmy, że nasza funkcja rekurencyjna daje się zapisać tak:

f(0) = 0
f(1) = 0 + 1 = 1
f(2) = 0 + 1 + 2 = 3
f(3) = 0 + 1 + 2 + 3 = 6
f(4) = 0 + 1 + 2 + 3 + 4 = 10
f(5) = 0 + 1 + 2 + 3 + 4 + 5 = 15
f(6) = 0 + 1 + 2 + 3 + 4 + 5 + 6 = 21
...
f(n) = 0 + 1 + 2 + 3 + ... + n = ?
Czy zauważasz już pewną prawidłowość? Czy to co zapisałem wyżej czegoś nie przypomina? TAK - jest to suma n pierwszych liczb naturalnych, ale zaraz zaraz - czy nie znasz może wzoru na taką sumę? Na pewno znasz, tylko nie pamiętasz, ale nie ma problemu ja go tu przypomnę:

$$f(n) = \frac{n(n+1)}{2}$$

I wykorzystując ten prosty wzór można zobaczyć, że program wykonuje się szybciej.