

41. Desenvolver um programa em Prolog, que apresente o valor de g utilizando a série

$$g = 1/1! - 2/1! + 3/2! - 4/3! + 5/5! - 6/8! + 7/13! - \dots + n/F_n!$$

Os termos da sequência de Fibonacci, são definidos recursivamente pela fórmula $F_n = F_{n-1} + F_{n-2}$, sendo $F_1 = 1$ e $F_2 = 1$.

O valor de n será fornecido pelo usuário, devendo ser um valor inteiro maior do que zero.

Por exemplo, caso o valor fornecido pelo usuário para n seja 5, o programa deverá apresentar como resposta o valor -0.12499999999999997 , ou seja, $1/1! - 2/1! + 3/2! - 4/3! + 5/5!$.

Caso o usuário forneça um valor inválido para n , o programa deverá apresentar como resposta o valor `false`.

```
% Função fatorial
fatorial(0, 1) :- !.
fatorial(N, F) :- K is N - 1, fatorial(K, G), F is G * N.

% Função fibonacci
fibonacci(1, 1) :- !.
fibonacci(2, 1) :- !.
fibonacci(N, R) :- N1 is N - 1, N2 is N - 2, fibonacci(N1, F1), fibonacci(N2, F2),
    R is F1 + F2.

% Função série G
serieG(N, false) :- N < 1, !.
serieG(1, 1) :- !.
serieG(N, G) :- N mod 2 == 0, K is N - 1, serieG(K, R), !, fibonacci(N, T),
    fatorial(T, F), X is N / F, G is R - X.
serieG(N, G) :- N mod 2 == 1, K is N - 1, serieG(K, R), !, fibonacci(N, T),
    fatorial(T, F), X is N / F, G is R + X.

% Execução
?- serieG(0, G), write(G), nl. % false
?- serieG(1, G), write(G), nl. % 1
?- serieG(2, G), write(G), nl. % -1
?- serieG(3, G), write(G), nl. % 0.5
?- serieG(4, G), write(G), nl. % -0.16666666666666663
?- serieG(5, G), write(G), nl. % -0.12499999999999997
?- serieG(6, G), write(G), nl. % -0.1251488095238095
?- serieG(7, G), write(G), nl. % -0.12514880839967643
?- serieG(8, G), write(G), nl. % -0.12514880839967643
```