

Implemente os seguintes predicados:

1. `sequencia(A, B, L)` - sucede quando L é a lista com uma sequência de números entre A e B.

$L = [A, A+1, \dots, B-1, B]$

2. `double(L1, L2)` - sucede quando L2 é uma versão de L1 com os elementos todos "duplicados".

ex: $L1 = [1, 2, 3]$, $L2 = [1, 1, 2, 2, 3, 3]$

3. `adj(E1, E2, L)` - sucede quando E1 e E2 estão em posições adjacentes na lista L

ex: `adj(1, 2, [1, 2, 3, 4, 5])` ou `adj(1, 2, [3, 2, 1, 4, 5])` são ambas verdadeiras

4. `sel(X, L1, L2)` - sucede quando L2 é o resultado da remoção do elemento X da lista L1 (igual ao predicado `select/3` do Prolog)

5. `somatorio(L, S)` - sucede quando S é o somatório dos elementos de L

6. `ord(L)` - sucede quando todos os elementos da lista L estão ordenados

7. `perm(L1, L2)` - sucede quando L2 é uma permutação da lista L1

nota: pode (e deve) utilizar o predicado `sel/3` para ajudar

TPC

Consultando os slides das aulas teóricas, implemente o seguinte predicado:

8. `sort(L1, L2)` - sucede quando L2 é uma ordenação da lista L1

Última alteração: terça, 4 de outubro de 2022 às 11:33

 [Contactar suporte do site](#) 

Nome de utilizador: [Rodrigo Alves](#) ([Sair](#))

[Resumo da retenção de dados](#)

[Obter a Aplicação móvel](#)

Fornecido por [Moodle](#)

