

## Banco de questões para a prova de prolog (2024.1)

### Questões: Introdução à Prolog

#### 1 - Entre com um número e retorne se ele é maior ou menor que 100:

```
maiorQueCem(X) :- write(Entre com o número:'),
read(X),
(
    (X>100, write('O número é maior que cem'));
    (X<100, write('O número é menor que cem'))
).
```

**2 - Entre com uma nota de aluno e descubra se ele foi aprovado, reprovado ou está em recuperação, lembrando que o máximo que o aluno pode tirar é 10. Se a sua nota for maior ou igual a 7 e menor que 10, ele está aprovado, se a nota for maior ou 5 e menor ou igual a 6.9, ele está em recuperação e se a nota for menor que 4.9, ele está reprovado.**

```
situacaoAluno (Aluno):- nota(Aluno, Nota), (( Nota >= 7.0 , Nota < 10.0,
write('Aprovado'));
( Nota >= 5.0, Nota < 6.9, write('Recuperação'));
( Nota >= 0.0, Nota < 4.9, write('Reprovado')));).
```

### Questões: Recursão

#### 1 - Encontrar fatorial:

```
fatorial(0, 1).
fatorial(N, F) :- N > 0,
    N1 is N - 1,
    fatorial(N1, F1),
    F is N * F1.
```

### Questões: Conjuntos

### Questões: Listas

#### 1 - Saber se um número é membro de uma lista:

```
membro(X, [X|_]).
membro(X, [_|R]) :- membro(X,R).
```

#### 2 - Escreva as regras que removem elementos de uma lista (caso remover)

```
remove(X, [X|C], C).
remove(X, [Y|Z], [Y|P]) :- remove(X, Z, P).
```

OU

```
remover(X, [X], []).
remover(X, [X|T], T).
remover(X, [Y|W], [Y|Z]) :- remover(X, W, Z).
```

**3 - Exercício do quadrado menor:**

quadradoMenor(\_,[],[]) :- !.  
quadradoMenor(X,[Y|R],[Y|W]) :- C is Y\*Y,  
C < X,  
quadradoMenor(X,R,W), !.

quadradoMenor(X,[\_R], W) :- quadradoMenor(X,R,W).

**4 - Insere elementos em uma lista:**

insereElemento(E, [], [E]) :- !.  
insereElemento(E, [X|R],[E,X|R]).  
insereElemento(E, [X|R],[X|W]) :- insereElemento(E,R,W).

**5 - Conta quantos elementos existem em uma lista:**

contagem([], 0).  
contagem(\_|T, N) :- contagem(T, N1), N is N1+1.

**6 - Soma todos os elementos de uma lista:**

soma([], 0).  
soma([X|W], S):- soma(W, S1), S is S1 + X.

**7 - Multiplicação de todos os elementos da lista:**

multiplicacao([], 0).  
multiplicacao(L, P) :- produto(L,P) .  
  
produto([], 1).  
produto([X|W], S) :- produto(W, S1), S is S1 \* X.

**8 - Obter o primeiro elemento de uma lista:**

p\_elemento([X|T], X).

**9 - Obter o último elemento de uma lista:**

u\_elemento([X], X) :- !. u\_elemento(\_|T, L) :- u\_elemento(T, L).

**10 - Mergear 02 listas distintas:**

merge([], [], []).  
merge(H, [], H).  
merge([], H, H).  
merge([H|T], L, [H|Z]) :- merge(T, L, Z).

**11 - Mergear e ordenar 02 listas distintas:**

