

Primeira lista de exercícios

Listas em Prolog

Exercício 1:

Escreva o predicado Prolog `analisa_lista/1` que toma uma lista como argumento e escreve sua cabeça e cauda na tela. Se a lista está vazia, o predicado deve escrever uma mensagem. Exemplos de uso:

```
?- analisa_lista([dog, cat, horse, cow]).  
A cabeça da lista eh: dog  
A cauda da lista eh: [cat, horse, cow]  
True  
?- analisa_lista([]).  
A lista esta vazia  
True
```

Exercício 2:

Implemente o predicado Prolog `remove_duplicados/2` que remove todos os elementos duplicados de uma lista dada como primeiro argumento e retorna o resultado no segundo argumento. Exemplo de uso:

```
?- remove_duplicados([a, b, a, c, d, d], List).  
List = [b, a, c, d]  
True
```

Exercício 3:

Escreva o predicado Prolog `troca/4` que troca todas as ocorrências de um dado elemento (segundo argumento), por outro elemento (terceiro argumento) de uma lista (primeiro argumento). Exemplo de uso:

```
?- troca([1, 2, 3, 4, 3, 5, 6, 3], 3, x, List).  
List = [1, 2, x, 4, x, 5, 6, x]  
True
```

Exercício 4:

Uma lista Prolog sem valores duplicados pode ser usada para representar um conjunto. Escreva um predicado que dado uma lista que representa um conjunto C, produza uma lista de lista que é o conjunto potência de C. Exemplo de uso:

```
?- potencia([a, b, c], P).  
P = [[a, b, c], [a, b], [a, c], [a], [b, c], [b], [c], []]  
True
```

Exercício 5:

Escreva o predicado Prolog `mais_longa/2` que recebe duas listas como argumentos e verifica se a segunda lista é mais longa (tem mais elementos) que a primeira lista. Implementar sem usar qualquer operação aritmética (contar o tamanho das listas, por exemplo). Exemplo de uso:

```
?- mais_longa([dog, cat, snake], [giraffe, elephant, lion, tiger]).  
True  
?- mais_longa([1, 2, 3, 4, 5], []).  
False
```

Exercício 6:

Escreva o predicado Prolog `distancia/3` que calcula a distância entre dois pontos no espaço bidimensional. Os pontos são dados como um par de coordenadas. Exemplo de uso:

```
?- distancia((0,0), (3,4), X).
X = 5.0
True
?- distancia((-2.5,1), (3.5,-4), X).
X = 7.810249675906654
True
```

Exercício 7:

Escreva o predicado Prolog `quadrado/2` para escrever um quadrado $n \times n$ de um dado caracter na tela. O primeiro argumento é o valor de n e o segundo o caracter a ser implementado. Exemplo de uso:

```
?- quadrado(5, '*').
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
True
```

Exercício 8:

Escreva o predicado Prolog `elemento_n/3`, que dado uma lista é um número natural n , retorna o n -ésimo elemento da lista. Exemplo de uso:

```
?- elemento_n([tiger, dog, teddy_bear, horse, cow], 3, X).
X = teddy_bear
True
?- elemento_n([a, b, c, d], 27, X).
False
```

Exercício 9:

Escreva o predicado Prolog `media/2` que calcula a média de uma lista de inteiros. Exemplo de uso:

```
?- media([1, 2, 3, 4], X).
X = 2.5
True
```

Exercício 10:

Escreva o predicado Prolog `minimo/2` que encontra o menor valor numérico de uma dada lista de números. Exemplo de uso:

```
?- minimo([4, 6, 8, 3, 5, 7], Result).
Result = 3
True
```

Exercício 11:

Escreva o predicado Prolog `intervalo/3` que gera uma lista de inteiros dado o limite inferior e superior como primeiro e segundo argumento. O resultado é uma lista de inteiros retornado como terceiro argumento. Exemplos de uso:

```
?- intervalo(3, 11, X).
```

```
X = [3, 4, 5, 6, 7, 8, 9, 10, 11]
True
?- intervalo(7, 4, X).
X = []
True
```

Exercício 12:

Implemente o algoritmo de Euclides para calcular o Máximo Divisor Comum de dois números inteiros não negativos. Exemplo de uso:

```
?- mdc(57, 27, X).
X = 3
True
```

Exercício 13:

Implemente o predicado Prolog `ocorrencias/3` que conta o número de ocorrências de um dado elemento numa lista. Exemplo de uso:

```
?- ocorrencias(dog, [dog, frog, cat, dog, dog, tiger], N).
N = 3
True
```

Exercício 14:

Escreva o predicado Prolog `divisores/2` que calcula a lista de divisores de um número. Exemplo de uso:

```
?- divisores(30, X).
X = [1, 2, 3, 5, 6, 10, 15, 30]
True
```