

## Instruções para Tarefa 04 – PLIA – EC/UFU (Valor 40 pontos)

Data/horário limite para entregar a tarefa: 23/12/2020 (23h59min50s)

Procedimentos para entregar:

Enviar um e-mail para [marcelo.ufu@gmail.com](mailto:marcelo.ufu@gmail.com) com as seguintes características:

- No campo assunto, coloque PLIA-2020-T4-<sua matrícula>.  
Exemplo PLIA-2020-T4-11911ECP999.
- Coloque nos anexos todo o código fonte feito por você, lembre-se de colocar seu nome e matrícula no arquivo.
- Faça o exercício 4 e implemente em PROLOG, à sua escolha, pelo menos 6 exercícios da lista abaixo (o exercício 4 não conta!).

## Lista de Exercícios

(Obs.: nos predicados "+" significa entrada e "-" significa saída)

1) Escreva um programa Prolog que faça o produto interno entre duas listas do mesmo tamanho, exemplo:

?- produto\_interno([1,2,3],[-1,2,0],X).     X = 3 (resultado da soma 1.-1+2.2+3.0).

2) Faça dois predicados Prolog que :

- bin\_para\_octal(Bin,Octal) – transforma uma lista que representa um número binário em outra lista que representa um número equivalente em octal.

- octal\_para\_bin(Octal,Bin) – transforma uma lista que representa um número octal em outra lista que representa um número equivalente em binário.

Exemplo:                      ?- bin\_para\_octal([1,1,1,1,0,1,0],X).                      X = [1,7,2].  
                                    ?- octal\_para\_bin([7,3,5]).                                X = [1,1,1,1,1,0,1,0,1]

3) N tarefas devem ser realizadas por N diferentes máquinas. O custo do uso de cada máquina i para realizar a tarefa j é determinado pela tabela:

		Tarefas			
		1	2	3	4
Máquinas	1	6	5	3	6
	2	5	6	8	12
	3	8	6	8	9
	4	3	6	5	8

Escreva um programa Prolog que determine quais tarefas devem ser atribuídas a cada máquina de tal forma a minimizar o custo total de suas realizações.

4) Preencha a tabela abaixo com as soluções às questões Prolog:

Questões		Solução	true/false
01	?- [ H   T ] = [ [ [ ] ] ].		
02	?- [ H   T ] = [ [ a, b ] ].		

03	?- [ H1, H2   T ] = [ [ a, b ] ].	
01	?- [ H1, H2   T ] = [ [ ]   [ a ] ].	
05	?- [ _   T ] = [ a, b   [ c ] ].	
06	?- [ H   _ ] = [ [a]   b, c ].	
07	?- [ [ 1 ], X   X ] = [ X, Y, Z ].	
08	?- [ X   X ] = [ [ ] ].	
09	?- [ X   X ] = [ [ ], [ ] ].	
10	?- [ X, X ] = [ [ ], [ ] ].	
11	?- [ X, Y, _   Z ] = [ 1, 2, 3, 4 ].	
12	?- [ X   [ _ , Z ] ] = [ 1, 2, 3, 4 ] .	
13	?- [ X   [ Y   Z ] ] = [ 1, 2, 3, 4 ] ..	
14	?- [ 1   [ Y, X ] ] = [ X, 2, 3 ].	
15	?- 2 - 2 + C = A - B + 5	
16	?- 3 + 0 = 4 - 1	
17	?- 3 + 02 =:= 4 - 1	
18	?- pai(joao,X) = pai(Y,maria)	
19	?-pai(Joao,X) = pai(adao,ada)	
20	?- 1 + 2 is 3	

5) Seja a árvore genealógica a seguir, determinada pelos seguintes fatos Prolog:

filho(júnior, josé).	filha(ana, josé).	homem(josé).	mulher(ana).
filho(júnior, ada).	filha(ana, ada).	homem(júnior).	mulher(ada).
filho(marcos, mário).	filha(maria, josé).	homem(marcos).	mulher(maria).
filho(marcos, ana).	filha(maria, ada).	homem(márcio).	mulher(alana).
filho(márcio, mário).	filha(alana, júnior).		mulher(cláudia).
filho(márcio, ana).	filha(alana, cláudia).		

A partir desses, construa regras Prolog que definam os relacionamentos:

- pais(X,Y) – X é pai ou mãe (um dos pais) de Y.
- irmão(X,Y) – X é irmão de Y.
- primos(X,Y) – X e Y são primos.
- tia(X,Y) – X é tia de Y.
- sobrinho(X,Y) – X é sobrinho de Y.
- bisavó(X,Y) – X é bisavó de Y.
- ascendente(X,Y) – X é descendente de Y.

6) Seja o seguinte programa PROLOG

g([ ], [ ]).

$g([H | T], [H | R]) :- 0 \text{ is } H \bmod 2, g(T, R).$

$g([H | T], R) :- 1 \text{ is } H \bmod 2, g(T, R).$

Qual a resposta deste programa à questão  $g([2, 1, 5, 4], Z)$ ?

O que este programa faz?

Justifique sua resposta através de um *trace* da questão.

7) Construa um predicado Prolog `divideN(+Lista, +N, -ListaN, -ListaR)` que dada uma Lista e um valor N separe a lista em duas listas, contendo a primeira (ListaN) os N primeiros elementos da lista original e a segunda (ListaR), os restantes elementos. Exemplos:

?- `divideN([5,4,1,3,2,6], 4, ListaN, ListaR).`

ListaN = [5,4,1,3] % 4 primeiros elementos

ListaR = [2,6] % Restantes elementos

?- `divideN([5,4,1,3,2,6], 3, ListaN, ListaR).`

ListaN = [5,4,1] % 3 primeiros elementos

ListaR = [3,2,6] % Restantes elementos

8) Pretende-se definir em Prolog um predicado `primos(+N, -Lista)` para gerar em Lista a sequência de números primos até N. O algoritmo a ser usado baseia-se na geração de uma lista de inteiros de 2 a N e na sucessiva remoção desta lista dos números que sejam múltiplos dos primos já identificados.

Para N=100, por exemplo, a lista inicial seria:

[2, 3, 4, 5, ..., 98, 99, 100]. Nesta lista o primeiro número que encontramos é 2, que é um primo. A partir do número seguinte podem remover-se todos os múltiplos de 2: 4, 6, 8, ..., 100. 3 é agora o número a considerar, e é o primo seguinte. A partir deste podemos remover os seus múltiplos: 9, 15, ..., 99. Repetindo este processo a lista resultante tem todos os primos de 2 a 100 e só esses.

a) Escreva um predicado para gerar uma lista com todos os números naturais desde 2 até ao limite desejado (N). Predicado: `ints(N, ListaInts)`

b) Escreva um predicado para remover de uma lista dada todos os múltiplos de um número K. Predicado: `removeMult(+K, +Lista, -NovaLista)`

c) Escreva o predicado `primos(+N, -Lista)`, usando os predicados anteriores. Exemplo:

?- `primos(12, Lista).`

Lista = [2, 3, 5, 7, 11]

9) Construa um predicado `extremos(+Lista, -Maximo, -Minimo)` que dada uma lista, determine os seus valores máximo e mínimo. Exemplo:

?- `extremos([5, 10, 2, 30, 6], Max, Min).`

Max = 30, Min = 2

10) Construa um predicado que rode de N ( $N \geq 0$ ) elementos para a esquerda uma lista. Predicado `rodar(+Lista, +N, -ListaRodada)`. Exemplo:

?- `rodar([a,b,c,d,e,f,g,h], 3, X).`

X = [d,e,f,g,h,a,b,c]

11) Escreva um predicado Prolog múltiplo/1 que recebe uma lista L e verifica se existe uma múltipla ocorrência de algum elemento. Exemplo: `?- múltiplo([a,b,c,b]).`

`true`

12) Escreva um predicado Prolog meio/2 que, dada uma lista, retorna o elemento que ocupa a posição do meio dessa lista. Se a lista tiver comprimento par o predicado deve falhar.

Exemplo: `?- meio([a,b,c,d,e],X).`

`X = c`

`true`

`?- meio([a,b,d,e],X).`

`false`

13) Considere a função f definida como:

$f(x) = x - 10$ , se  $x > 100$ ,

$f(x) = f(f(x+11))$ , se  $x \leq 100$ .

a) Escreva um Predicado Prolog  $f(X,Y)$  que define Y como o valor da função f para qualquer inteiro X.

b) Determine a resposta do seu predicado à questão: `?- f(95,Y).`

14) Faça um predicado Prolog que transforma graus Celsius em Fahrenheit.

Obs.:  $C / 100 = (F-32) / 180$ , o functor do predicado deverá ser transform e sua aridade é dois.

15) Para a determinação da raiz quadrada de um número real x,  $0 < x < 2$ , podemos utilizar o seguinte algoritmo:

Sejam  $a_0 = x$ ,  $c_0 = 1 - x$  e, para  $i > 0$ ,

$$a_i = a_{i-1} * (1 + c_{i-1}/2)$$

$$c_i = (c_{i-1})^2 * (3 + c_{i-1})/4$$

Pode-se demonstrar que o limite de  $a_n$  com n tendendo ao infinito é a raiz quadrada de N.

Faça um programa Prolog que determine a raiz quadrada de um número real no intervalo acima, com pelo menos 5 dígitos significativos. (Dica: para tal verifique se  $|a_i - a_{i-1}| < 0.0000001$ )

Considerações:

- Existe uma função nativa `abs(X)` que retorna o módulo de um número X;
- O programa deve ter o predicado principal de functor `prog1` e aridade 2, com os parâmetros X e RaizX.

16) Sejam L, L1 e L3 listas de termos. Escreva um programa Prolog que:

a) delete o n-ésimo elemento em L, sendo L1 a lista que sobra desta operação.

b) intercale elementos de L1 e L2 em L; por exemplo: se  $L1 = [a, b, c]$  e  $L2 = [1, 2]$ , então  $L = [a, 1, b, 2, c]$ . O seu programa funciona com questões inversas?

c) transponha L1 e L2 em L, isto é, se L1 = [a, b, c] e L2 = [1, 2, 3], então L = [(a,1),(b,2),(c,3)] . O seu programa funciona com questões inversas?

17) Faça um programa Prolog que transforma um número binário para hexadecimal e vice-versa (predicado transf\_bh(Binario,Hexadecimal), os números estarão representados por uma lista.

Exemplo: ?- transf\_bh([1,1,1,1,0,1,0],X). X = [7,A].  
?- transf\_bh(X,[A,F]). X = [1,1,1,1,1,0,1,0]

18) Faça um programa Prolog que faz o extenso de um valor em reais. Exemplo: de R\$ 1.234.459,90 temos “um milhão duzentos e trinta e quatro mil quatrocentos e cinquenta e nove reais e noventa centavos”.

19) Faça um programa que reverte uma lista. Predicado reverso(Lista,Atsil)

Exemplo: ?-reverso([r,o,m,a],X) X = [a, m, o, r]

20) Utilize o predicado reverso e faça um programa que determina se uma lista é palindromo. Predicado palindrome(Lista)

Exemplo: ?-palindrome([a,r,a,r,a]) true

21) Uma companhia está construindo uma nova sede com 3 andares. O primeiro andar tem 50 escritórios, o segundo tem 40 e o terceiro 30. A companhia tem 10 divisões para as quais requesita-se:

Divisão	dhw	dms	dsw	eme	epr	esn	mem	mes	mex	dcr
Número de Escritórios	10	22	08	14	11	09	09	09	16	12

Todos os escritórios de uma divisão devem ficar em um mesmo andar. Como também, os seguintes pares de divisões que tem serviços complementares deve ficar em um mesmo andar: dcr e dsw, dms e esn, eme e mex. Faça um programa Prolog que determina quais divisões devem ser alocadas em cada andar.