| MATA56 - Paradigmas de Linguagens de Programação  Prova 1  Aluno: |
| --- |

1. Você tem 110 minutos para fazer esta prova.

2. Você pode consultar qualquer material que estiver com você, mas não pode acessar a Internet ou consultar qualquer fonte externa, inclusive o seu colega ☺.

3. Você pode usar computador, mas não pode usar nenhum mecanismo de comunicação de dados do computador. 4. A codificação Cn, usada no início dos enunciados, indica quantas cláusulas eu usei para solucionar uma dada questão. Por exemplo: C2 indica que eu resolvi uma dada questão utilizando apenas duas cláusulas.

5. Você só pode usar cláusulas pré-definidas quando estas forem explicitamente mencionadas nos enunciados das questões, qualquer outra cláusula auxiliar deve ser codificada por você.

6. Assine a prova e as folhas de resposta logo no início do exame (i.e., AGORA mesmo).

7. Coloque a caneta de lado assim que o professor anunciar o final da prova.

**Questão 1 (2 pontos) - C2:** Escreva em PROLOG as cláusulas juntar(+L1,+L2,?L3) para juntar duas listas, intercalando seus elementos, como pedido na Questão 4 da Lista 1. Por exemplo:

?- juntar([a,b,c],[d,e,f,g,h], L).

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

IMPORTANTE: **você só deve utilizar somente duas cláusulas** para resolver esta questão. **Soluções com mais de duas cláusulas valem zero**.

DICA: inverta a posição de L1 e L2 na chamada recursiva.

**juntar([], X, X).**

**juntar([X|R], [Z|Q], [X,Z|W]) :- juntar(R, Q, W).**

**Questão 2 (2 pontos) - C2:** Escreva em PROLOG as cláusulas divide(+L,+N,?L1,?L2)que divide a lista L nas listas L1 e L2, sendo que os N primeiros elementos de L estarão em L1 e o restante em L2. Por exemplo:

?- divide([a,b,c,d,e,f,g,h,i,k],3,L1,L2).

L1 = [a,b,c]

L2 = [d,e,f,g,h,i,k]

As cláusulas devem falhar se N for maior que o comprimento de L.

**divide([X|R], 1, [X], R).**

**divide([X|R], N, [X|L], Z) :- N > 1, N1 is N - 1, divide(R, N1, L, Z)**.

**Questão 3 (2 pontos) - C5:** Escreva em PROLOG as cláusulas empacote(+L,?LL), que transforma uma lista em uma lista de listas, empacotando elementos iguais em sublistas distintas. Por exemplo:

?- empacote([a,a,a,b,c,c,a,a,d,e,e,e,e],X).

X = [[a,a,a,a,a],[b],[c,c],[d],[e,e,e,e]]

**empacote([], []).**

**empacote([X], [[X]]).**

**empacote([X|R], [Y|W]) :-**

**empacotarIguais(X, R, Y, Rest),**

**empacote(Rest, W).**

**empacotarIguais(X, [], [X], []).**

**empacotarIguais(X, [Y|T], [X|W], L) :-**

**X = Y,**

**empacotarIguais(X, T, W, L).**

**empacotarIguais(X, [Y|T], [X], [Y|T]) :-**

**X \= Y.**

**Questão 4 (2 pontos) - C3:** Escreva em PROLOG as cláusulas codifique(+L,?LL), que codifica a lista L em uma lista de listas LL codificada por comprimento linear dos elementos repetidos em L. Por exemplo:

?- codifique([a,a,a,b,c,c,a,a,d,e,e,e,e],X).

X = [[5,a],[1,b],[2,c],[1,d],[4,e]]

Você pode assumir a existência das cláusulas empacote da questão anterior e length/2 do Prolog.

**empacotarIguais(X, [], [X], []).**

**empacotarIguais(X, [Y|T], [X|W], L) :-**

**X = Y,**

**empacotarIguais(X, T, W, L).**

**empacotarIguais(X, [Y|T], [X], [Y|T]) :-**

**X \= Y.**

**empacote([], []).**

**empacote([X], [[X]]).**

**empacote([X|R], [Y|W]) :-**

**empacotarIguais(X, R, Y, Rest),**

**empacote(Rest, W).**

**codificaLista([], []).**

**codificaLista([[X|T]|W], [[N, X]|Z]) :-**

**length([X|T], N),**

**codificaLista(W, Z).**

**codifique(L, LL) :-**

**empacote(L, X),**

**codificaLista(X, LL).**

**Questão 5 (2 pontos) - C5:** Escreva em PROLOG as cláusulas circule(+L,+N,?LR), que recebe uma lista L e gira ciclicamente os seus elementos N vezes para esquerda, se N>0 e para direita, se N<0 . Por exemplo:

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

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

?- circule([a,b,c,d,e,f,g,h],-2,X).

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

Você pode assumir a existência das cláusulas append/3 do PROLOG.

**juntar([], L, L).**

**juntar([X|L1], L2, [X|L3]) :- juntar(L2, L1, L3).**

**divide(L, 0, [], L).**

**divide([X|L], N, [X|L1], L2) :-**

**M is N-1,**

**divide(L, M, L1, L2).**

**separa([X], [X], []).**

**separa([X|L], [X,X|L1], L2) :- separa(L, [X|L1], L2).**

**separa([X|L], L1, [X|L2]) :- separa(L, L1, L2).**

**empacote([], []).**

**empacote(L, [LS|LL]) :-**

**separa(L, LS, LR),**

**empacote(LR, LL).**

**codifique([], []).**

**codifique(L, [[X,N]|LL]) :-**

**separa(L, [X|LS], LR),**

**length([X|LS], N),**

**codifique(LR, LL).**

**circule(L, 0, L).**

**circule(L, N, LO) :-**

**N > 0,**

**M is N-1,**

**append(R, [X], LO),**

**circule(L, M, [X|R]).**

**circule(L, N, [X|LO]) :-**

**N < 0,**

**M is N+1,**

**append(LO, [X], R),**

**circule(L, M, R).**