

**Identificação do aluno:**

Nome: \_\_\_\_\_ #: \_\_\_\_\_

**I**

1. Analise os predicados escritos em Mercury apresentados em seguida e explique o que fazem:

- a)     :- pred primeiro(list(int),int).  
       :- mode primeiro(in, out) is det.  
       primeiro( [],0).  
       primeiro([A|B],D)  
       :- primeiro(B,C),  
          ( if A>0 then D = A+C else D = C ).

- b)     :- pred segundo(list(T)::in,list(T)::out,list(T)::out) is multi.  
       segundo(A,[],A).  
       segundo([A|B],[A|C],D)  
       :- segundo(B,C,D).

- c)     :- pred terceiro( {list(T1),list(T2),list(T3)}::in, list({T1,T2,T3})::out ) is det.  
       terceiro({A,B,C},D)  
       :- E = ( pred({F,G,H}::out) is nondet :- member(F,A),member(G,B),member(H,C) ),  
          solutions(E,D).

2. O módulo apresentado a seguir define um conjunto de tipos para representar um vendedor retalhista de peças Lego, bem como um conjunto de predicados e funções para manipular esses tipos. O objectivo do módulo é gerir o catálogo de peças: as operações definidas permitem acrescentar ou remover peças bem como obter informação sobre a quantidade existente.

As peças, representadas pelo tipo **part**, têm as seguintes características:

- Código numérico.
- Cor – representada pelo tipo **part\_colour**. As cores existentes são o branco, azul, amarelo, vermelho, verde e preto.
- Forma – representado pelo tipo **part\_shape**. Todas as peças têm uma forma de um tijolo, apenas variando o comprimento e a largura (a altura é sempre de 1 unidade). Assim, Brick 1x2 representa um tijolo de largura 1 e comprimento 2.

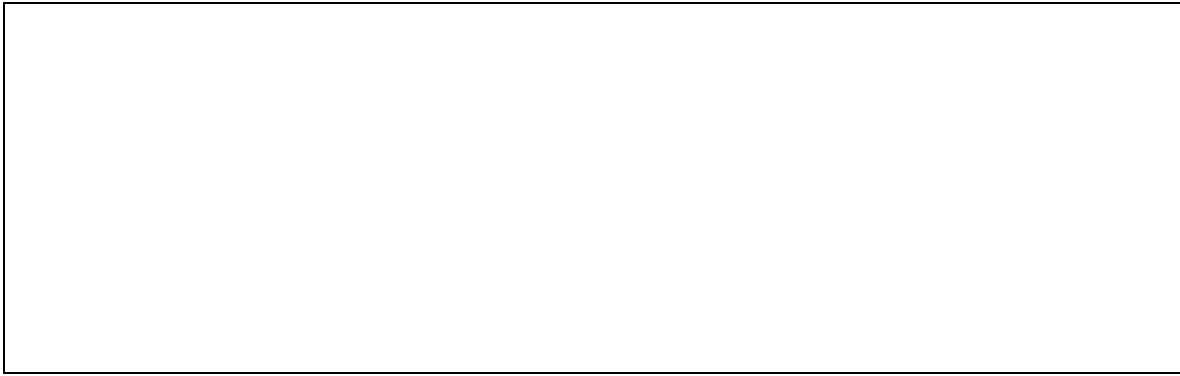
Para além da relação das peças, a um catálogo (tipo **catalog**) está associado o nome da pessoa que o gere. A tabela aqui apresentada representa um possível catálogo. O tipo **stock** representa tudo o que existe em armazém e consiste de um conjunto de catálogos.

Retalhista: Jorge Pires			
Cod	Forma	Cor	Quantidade
1	Brick 2x2	Amarelo	50
2	Brick 2x4	Azul	99
3	Brick 2x2	Verde	27
4	Brick 2x6	Amarelo	38

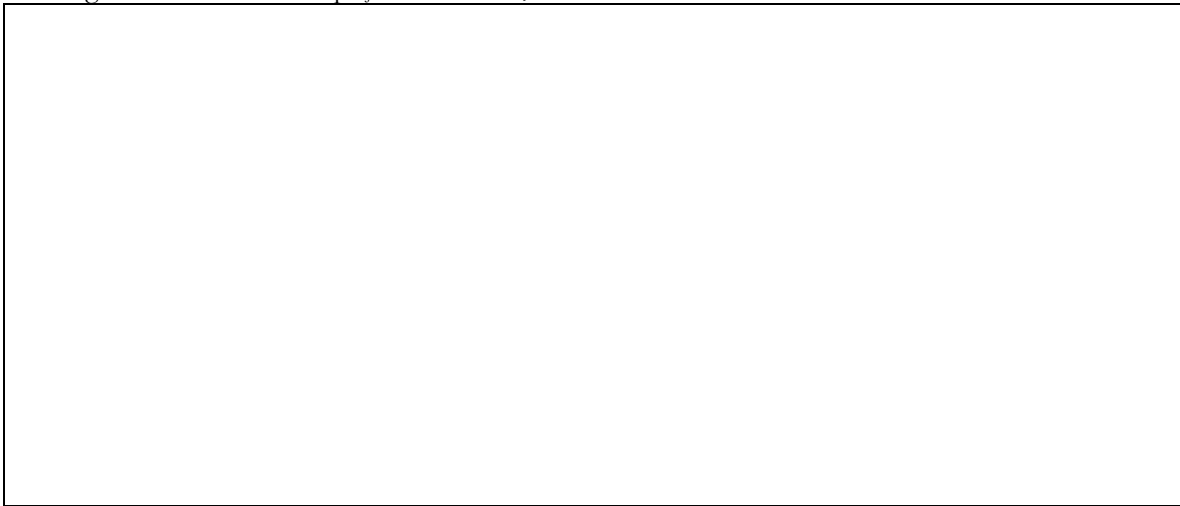
Considere o seguinte esqueleto do módulo que define as funcionalidades pretendidas:

<pre>:- module lego_stock. %%%%%%%%%%%%%%%%%%  :- interface. %%%%%%%%%%%%%%%%%%  :- type stock.  :- type catalog.  :- type part.  :- type part_colour.  :- type part_shape.  :- func init_catalog(     string, % retailer name ) = catalog.</pre>	<pre>:- pred add(     catalog::in,     catalog::out,     part::in,     int::in) is det. % part quantity  :- pred remove(     catalog::in,     catalog::out,     part::in,     int::in) is det. % part quantity  :- pred color_quantity(     catalog::in,     part_color::in,     int::out) is det.  :- pred shape_quantity(     catalog::in,     part_shape::in,     int::out) is det.  :- implementation. %%%%%%%%%%%%%%%%%% ...</pre>
---	---

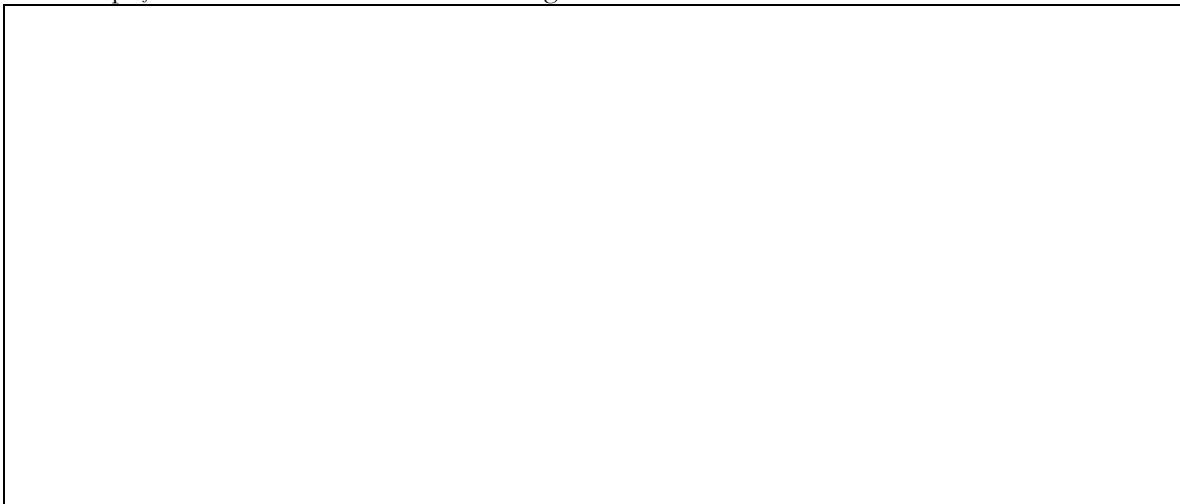
a) Defina os tipos **stock**, **catalog**, **part**, **part\_colour** e **part\_shape** e quaisquer outros tipos que julgue necessários à implementação dos predicados/funções pedidos nas alíneas seguintes.



b) Implemente o predicado `remove/4` que remove um determinado número de peças de um dado catálogo. Caso não existam peças suficientes, não remove nenhuma.



c) Implemente o predicado `colour_quantity/3` que, dado um catálogo e uma cor, devolve o número total de peças dessa cor existentes nesse catálogo.

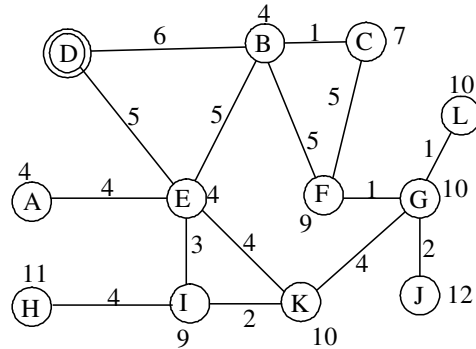


## II

1. Identifique semelhanças e diferenças entre a pesquisa em árvore em profundidade e a pesquisa por montanhismo.

2. As casas têm divisões de diferentes tipos, por exemplo, salas de estar, salas de jantar, quartos de dormir, cozinhas e quartos de banho. As divisões da casa tem peças de mobiliário, como por exemplo, mesas, cadeiras, camas, cómodas e estantes. A casa da Gabriela é em Aveiro. Essa casa tem um quarto com uma cama, em que a Gabriela dorme, e uma cómoda. Represente este conhecimento através de uma rede semântica.

3. O grafo a seguir apresentado representa um espaço de estados num problema de pesquisa, sendo **D** o estado objectivo (solução). As estimativas do custo de chegar à solução a partir de cada estado estão anotadas junto aos mesmos. Os custos das transições estão anotados junto às ligações do grafo.



a) Verifique se as estimativas de custo anotadas junto a cada nó constituem uma heurística admissível para a pesquisa A\*. Se não for esse o caso, introduza (na própria figura) alterações que a tornem admissível. Justifique.

b) Tomando o estado **G** como estado inicial, apresente a árvore de pesquisa gerada quando se realiza uma pesquisa A\* com repetição de estados. Numere os nós pela ordem em que são acrescentados à árvore e anote também o valor da função de avaliação em cada nó.

c) Indique um valor aproximado do factor de ramificação efectivo da árvore gerada.

4. Considere um veículo autónomo que se movimenta num ambiente estruturado em nós e ligações, ou seja, estruturado como um grafo. As ligações correspondem a ruas. Os nós podem conter cruzamentos de ruas e/ou parques de estacionamento. Os cruzamentos são locais em que convergem (começam ou terminam) quatro ruas. As ruas começam e terminam em nós adjacentes no grafo. O agente é capaz de realizar as seguintes acções: cruzar (seguir pela rua em frente), virar à esquerda, virar à direita, estacionar, continuar (seguir até ao nó no outro extremo da rua em que está), sair do estacionamento para uma dada rua (que começa ou termina no respectivo nó).

a) Identifique e caracterize um conjunto de predicados em lógica de primeira ordem que possam ser usados para especificar condições sobre estados de planeamento neste domínio. Identifique os valores possíveis dos argumentos desses predicados.

Nota: Para responder a esta pergunta, é aconselhável ver também a alínea b), onde estes predicados também são usados.

b) Usando os predicados que propôs, defina um conjunto de operadores STRIPS para representar as acções que podem ser realizadas neste domínio.

***Identificação do aluno:***

Nome: \_\_\_\_\_ #: \_\_\_\_\_

#INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

Exame, xx/xx/xxxx (Tempo: 3h)