

[Painel do utilizador](#)

As minhas unidades curriculares

[Programação em Lógica](#)

Provas

[Mini-Teste 2 Modelo](#)



Início	Segunda, 6 de Janeiro de 2020 às 16:05
Estado	Prova submetida
Data de submissão:	Segunda, 6 de Janeiro de 2020 às 19:04
Tempo gasto	2 horas 58 minutos

Pergunta 1

Respondida Pontuação 2,500

Explique o que faz o seguinte programa em Prolog e comente quanto à sua eficiência:

```
:- use_module(library(lists)).

p1(L1,L2) :-
    gen(L1,L2),
    test(L2).

gen([],[]).
gen(L1,[X|L2]) :-
    select(X,L1,L3),
    gen(L3,L2).

test([_,_]).
test([X1,X2,X3|Xs]) :-
    (X1 < X2, X2 < X3; X1 > X2, X2 > X3),
    test([X2,X3|Xs]).
```

% recebe 1 lista em L1 e gera para L2 uma lista igual à lista L1.
% depois disso, verifica que para cada grupo de 3 elementos consecutivos
% têm que estar ordenados por ordem crescente ou decrescente nao podendo haver elementos iguais

Pergunta 2

Respondida Pontuação 2,500

O Asdrúbal resolveu implementar uma versão equivalente usando programação em lógica com restrições (PLR):

```
:- use_module(library(clpfd)).

p2(L1,L2) :-
    length(L1,N),
    length(L2,N),
    %
    pos(L1,L2,Is),
    all_distinct(Is),
    %
    labeling([],Is),
    test(L2).

pos([],_,[]).
pos([X|Xs],L2,[I|Is]) :-
    nth1(I,L2,X),
    pos(Xs,L2,Is).
```

Contudo, após estudar melhor descobriu que o seu programa não era uma implementação correta em PLR. Escolha a opção que melhor traduz essa constatação:

Selecione uma opção de resposta:

- ☒ a. Não é possível etiquetar *Is*, pois estas variáveis não têm domínio definido, e nem todas as restrições foram colocadas antes da fase da pesquisa.
- ☐ b. Nem todas as restrições foram colocadas antes da fase da pesquisa.
- ☐ c. Não é possível etiquetar *Is*, pois estas variáveis não têm domínio definido.
- ☐ d. [Não Responder]
- ☐ e. As variáveis de domínio estão a ser instanciadas antes da fase de pesquisa.
- ☐ f. As variáveis de domínio estão a ser instanciadas antes da fase de pesquisa e nem todas as restrições foram colocadas antes da fase da pesquisa.

Pergunta 3

Respondida Pontuação 3,000

Corrija o programa, de modo a obter uma implementação correta em PLR.

```
p2(L1,L2) :-
    length(L1,N),
    length(L2,N),
    % generate list
    pos(L1, L2, Is),
    all_distinct(Is),
    %elimante symetry
    length(Aux, N), sorting(Is, Aux, Is),
    % check order
    testOpt(L2),
    % find solutions
    labeling([], Is).

pos([], _, []).
pos([X|Xs], L2, [I|Is]) :-
```

Pergunta 4

Respondida Pontuação 6,000

O Franquelim gosta de reciclar. Descobriu que tem na sua posse um determinado número de embalagens de iogurte vazias. Tem também um livro que mostra como realizar diversos objetos com tais embalagens, onde para cada objeto consta um custo estimado do material adicional que é preciso comprar (entre cola, tinta, etc.) e o número de embalagens de iogurte que leva. O Franquelim tem um orçamento limitado para comprar esses materiais, pretende realizar 3 objetos diferentes e utilizar o máximo possível de embalagens de iogurte de que dispõe. Usando programação em lógica com restrições, construa um programa que determine que objetos é que o Franquelim deve fazer. O predicado **build(+Budget,+NPacks,+ObjectCosts,+ObjectPacks,-Objects,-UsedPacks)** recebe o orçamento disponível (*Budget*), o número de embalagens de iogurte disponíveis (*NPacks*), os custos de material (*ObjectCosts*) e embalagens (*ObjectPacks*) necessários por cada objeto; devolve em *Objects* os objetos a construir (índices das listas *ObjectCosts/ObjectPacks*) e em *UsedPacks* o número de embalagens utilizadas.

```
| ?- build(60,30,[20,50,10,20,15],[6,4,12,20,6],Objects,UsedPacks).
Objects = [1,3,5],
UsedPacks = 24

| ?- build(120,30,[20,50,10,20,15],[6,4,12,20,6],Objects,UsedPacks).
Objects = [1,2,4],
UsedPacks = 30
```

```
build(Budget, NPacks, ObjectCosts, ObjectPacks, Objects, UsedPacks) :-
    length(ObjectCosts, N), length(ObjectPacks, N),
    length(Objects, 3), all_distinct(Objects),
    % restriction of the budget limit
    element(1, Objects, ObjIndex1), element(2, Objects, ObjIndex2), element(3, Objects, ObjIndex3),
    element(ObjIndex1, ObjectCosts, Cost1), element(ObjIndex2, ObjectCosts, Cost2), element(ObjIndex3, ObjectCosts, Cost3),
    Cost1 + Cost2 + Cost3 #< Budget,
    % % restriction of the packs limit
    element(1, Objects, ObjIndex1), element(2, Objects, ObjIndex2), element(3, Objects, ObjIndex3),
    element(ObjIndex1, ObjectPacks, Pack1), element(ObjIndex2, ObjectPacks, Pack2), element(ObjIndex3, ObjectPacks, Pack3),
    UsedPacks #= Pack1 + Pack2 + Pack3,
    UsedPacks #=< NPacks,
    % elimante symetry
    length(Aux, 3), sorting(Objects, Aux, Objects),
    % find solution
```

Pergunta 5

Respondida Pontuação 6,000

A Miquelina tem um conjunto de presentes para embrulhar. Tem igualmente diversos rolos de papel de embrulho, já abertos, com diferentes padrões mas de largura fixa. Para ser embrulhado, cada presente precisa de um determinado comprimento de papel. Construa um programa, usando programação em lógica com restrições, que determine com que rolo de papel é que cada presente deve ser embrulhado. O predicado **embrulha(+Rolos,+Presentes,-RolosSelecionados)** recebe a lista *Rolos* com a quantidade de papel disponível em cada rolo e a lista *Presentes* com a quantidade de papel que cada presente gasta; devolve em *RolosSelecionados* os rolos de papel a utilizar em cada presente.

```
| ?- embrulha([100,45,70], [12,50,14,8,10,90,24], S).
S = [2,3,3,2,1,1,2] ? ;
S = [3,3,2,3,1,1,2] ? ;
no
```

```
embrulha(Rolos, Presentes, RolosSelecionados) :-
    length(Rolos, Domain), length(Presentes, Length),
    length(RolosSelecionados, Length), domain(RolosSelecionados, 1, Domain),
    % presents restrictions
    presents(Rolos, Presentes, RolosSelecionados),
    labeling([], RolosSelecionados).

presents(Rolos, [], []).
presents(Rolos, [HPres|TPres], [HIndex|TIndex]) :-
    element(HIndex, Rolos, Quantity),
    Quantity #> HPres,
    NewQuantity #= Quantity - HPres,
    updateList(Rolos, HIndex, NewQuantity, NewRolos, 1),
    presents(NewRolos, TPres, TIndex).
```

