

[Painel do utilizador](#)[As minhas unidades curriculares](#)[Programação em Lógica](#)[Evaluation](#)[Mini-Teste 2 Modelo](#)**Início** quinta, 7 de janeiro de 2021 às 17:16**Estado** Prova submetida**Data de
submissão:** quinta, 7 de janeiro de 2021 às 17:18**Tempo gasto** 1 minuto 45 segundos

Pergunta 1

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

Pergunta 2

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

Pergunta 3

Não respondida

Pontuação 3,000

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

Pergunta 4

Não 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 ele precisa. 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
```

Pergunta 5

Não respondida

Pontuação 6,000

O Herberto 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 cada presente deve ser embrulhado. O predicado ***wrap(+Presents,+PaperRolls,-SelectedPaperRolls)*** recebe a lista de presentes *Presents* com a quantidade de papel que cada um gasta, e a lista de rolos *PaperRolls* com a quantidade de papel disponível em cada rolo; devolve em *SelectedPaperRolls* os rolos de papel a utilizar em cada presente.

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

[◀ TP2 - Entrega Final](#)[Ir para...](#)[SICStus Prolog ▶](#)