


[Minha página principal](#) ► [Programação em Lógica](#) ► [Provas](#) ► [Prova Modelo](#)

Informação


 Destacar
pergunta

Um grupo de pessoas vai fazer uma excursão. Pretende-se que as pessoas se conheçam melhor nas viagens (ida e volta). Para tal, em cada par de lugares da camioteta, nenhum par de pessoas se deve repetir na viagem de ida e de regresso. O número de lugares disponíveis corresponde sempre ao número de pessoas (em caso ímpar, uma das pessoas viajará sem par).

Pergunta **11**

Não respondida

Pontuação 2,00

 Destacar
pergunta

Pretende-se gerar uma lista de valores discretos para uma função da qual se conhece apenas o seu máximo global (maior valor possível) e o número de máximos locais ("picos" da função, i.e., valores que são mais altos do que os seus vizinhos). Implemente em PLR o programa **local_and_global_maxima(+NLocalMax,+GlobalMax,+N,-L)**, que gera os valores possíveis de uma função que tem **NLocalMax** máximos locais, e cujo máximo global (valor mais alto) é **GlobalMax**. Devem ser gerados **N** valores positivos, devolvidos na lista **L**.

```
| ?- L = [_X1,_X2,_,_], _X1 #< _X2, local_and_global_maxima(1,3,4,L).
```

```
L = [1,2,1,3] ? ;
```

```
L = [1,3,1,2] ? ;
```

```
L = [2,3,1,2] ? ;
```

```
no
```

```
| ?- local_and_global_maxima(2,3,5,L).
```

```
L = [2,1,2,1,3] ? ;
```

```
L = [2,1,3,1,2] ? ;
```

```
L = [3,1,2,1,2] ? ;
```

```
no
```

```
| ?- L = [_,_X2,_,_,_], maximum(_X2,L), local_and_global_maxima(1,3,5,L).
```

```
L = [1,3,1,1,2] ? ;
```

```
L = [1,3,1,2,1] ? ;
```

```
L = [1,3,2,1,2] ? ;
```

```
L = [2,3,1,1,2] ? ;
```

```
L = [2,3,1,2,1] ? ;
```

```
L = [2,3,2,1,2] ? ;
```

```
no
```

```
| ?- L = [_,_X2,_,_,_], maximum(_X2,L), local_and_global_maxima(2,3,5,L).
```

```
no
```

```
| ?- L = [_,_X2,_,_,_,_], maximum(_X2,L), local_and_global_maxima(2,3,6,L).
```

```
L = [1,3,1,2,1,2] ? ;
```

```
L = [2,3,1,2,1,2] ? ;
```

```
no
```

```
| ?- local_and_global_maxima(4,3,9,L).
```

```
L = [2,1,2,1,2,1,2,1,3] ? ;
```

```
L = [2,1,2,1,2,1,3,1,2] ? ;
```

```
L = [2,1,2,1,3,1,2,1,2] ? ;
```

```
L = [2,1,3,1,2,1,2,1,2] ? ;
```

```
L = [3,1,2,1,2,1,2,1,2] ? ;
```


```
no
```



Pergunta **12**

Não respondida

Pontuação 2,00

 Destacar
pergunta

Considere agora que cada fila do autocarro tem N lugares juntos. Generalize o programa anterior para que ninguém se reencontre na viagem de regresso (isto é, para cada duas pessoas que na ida vão na mesma fila de N lugares, têm que regressar em filas separadas).

Exemplos:

```
| ?- seatdown([a,b,c,d,e,f,g],3,Seats1,Seats2).  
no  
  
| ?- seatdown([a,b,c,d,e,f,g,h,i,j,k],3,Seats1,Seats2).  
[a,b,c]  
[d,e,f]  
[g,h,i]  
[j,k]  
  
[a,d,g]  
[b,e,h]  
[c,f,j]  
[i,k]  
  
Seats1 = [1,2,3,4,5,6,7,8,9,10|...],  
Seats2 = [1,4,7,2,5,8,3,6,10,9|...]  
  
| ?- seatdown([a,b,c,d,e,f,g,h,i,j,k],4,Seats1,Seats2).  
no
```

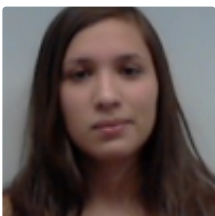
O conjunto de predicados seguinte permite obter o output apresentado:

```
show_seats(Ppl,N,Seats) :-  
    length(Row,N), append(Row,SeatsSufix,Seats),  
    !,  
    write_people(Ppl,Row),  
    show_seats(Ppl,N,SeatsSufix).  
show_seats(Ppl,_,Seats) :-  
    write_people(Ppl,Seats).  
  
write_people(Ppl,List) :-  
    get_ppl(Ppl,List,ListPpl), write(ListPpl), nl.  
  
get_ppl(_,[],[]).  
get_ppl(L,[X|Xs],[P|Ps]) :-  
    nth1(X,L,P),  
    get_ppl(L,Xs,Ps).
```



Seguinte

NAVEGAÇÃO NO TESTE



Marta Milheiro Soeiro Nunes Lopes

i 1 2 3 4 5 i 6 7 8 9 i 10 i 11 12 13

Terminar resolução...

Tempo restante **2:24:58**

© 2016 U.Porto - Tecnologias Educativas

Nome de utilizador: Marta Milheiro Soeiro Nunes Lopes. (Sair)

Gestão e manutenção da plataforma Moodle U.PORTO da responsabilidade da unidade de Tecnologias Educativas. Mais informações:

elearning@uporto.pt | +351 22 040 81 91 | <http://elearning.up.pt>



Based on an original theme created by Shaun Daubney | moodle.org