

✓ Concluída

1. Faça um programa que leia o standard input (user) e escreva no standard output (user, também) mas passando as maiúsculas para minúsculas.

O programa deverá ter como ativação o predicado `to_lc/0`, i.e. poderá ser usado com o goal:

```
| ?- see('xpti.txt'), tell('xpto.txt'), to_lc, seen, told.
```

2. Recupere as definições de **flatten/2** dadas na teórica T06 e aplique à listas, mas chame ao predicado **pferro/2**:

- `[1,[2,[3,[4],5],6],7]`
- `[[1,2], [3,4], [5,6]]`
- `[[1],[2,[3,4]], [5,[[6],7],8]`

Faça trace da execução do goal `pferro(L1, L2)` em que `L1` toma os valores acima.

3. Faça um predicado para ver se um string é uma capicua (**cap(S)** sucede se **S** for uma capicua)
4. Faça um comando "lsr(DIR)", em que lista recursivamente o conteúdo da diretoria passada como argumento. Deve apresentar:
 - o nome da diretoria
 - para cada ficheiro, o nome, o tipo e o tamanho em bytes
 - se o ficheiro for uma diretoria, deve chamar-se recursivamente com a nova (sub)diretoria
5. Faça um comando "find(NOME)" em que NOME é um nome de ficheiro; o objetivo vai ser encontrar a diretoria em que NOME se encontra, debaixo da diretoria atual.

Para os exercícios (4) e (5) pode usar os seguintes predicados de sistema (ver docs do Prolog):

- `working_directory(D)` ~ unifica `D` com a diretoria corrente
- `directory_files(D, Fs)` ~ unifica `Fs` com a lista de ficheiros na diretoria `D`
- `file_property(F, PROP)` ~ unifica `PROP` com uma propriedade do ficheiro `F`; pode ser:
 - `absolute_file_name(F)`
 - `real_file_name(F)`
 - `type(T)`**
 - `size(S)`**

Última alteração: terça, 18 de outubro de 2022 às 12:21

✉ [Contactar suporte do site](#) ↗

Nome de utilizador: [Rodrigo Alves](#) (Sair)

[Resumo da retenção de dados](#)

[Obter a Aplicação móvel](#)

Fornecido por [Moodle](#)

