



Exercícios #3

1 – Considera os predicados descritos na tabela, que não tens de definir

<i>receita/3</i>	<i>receita</i> (R, Item, Quant): A receita R necessita do Item na quantidade Quant.
<i>stock_minimo/2</i>	<i>stock_minimo</i> (Item, QuantMin): O stock mínimo do Item é a quantidade QuantMin

Escreve cláusulas Prolog que representem o conhecimento das seguintes regras informais:

Uma receita é não confeccionada se incluir items (ingredientes) cuja quantidade seja maior do que o stock mínimo existente.

Lista as receitas possíveis para a ementa do jantar.

Para cada receita indica quais os ingredientes em falta.

2 – Escreve o predicado *equal_sets/2* que tem sucesso se os dois conjuntos que receber são iguais. Nota que dois conjuntos iguais podem não ter os elementos pela mesma ordem.

Exemplos

```
?- equal_sets([a, b, c], [b, c, a]).  
true  
  
?- equal_sets([], []).  
true  
  
?- equal_sets([a, b, c, 1], [a, b, c]).  
false  
  
?- equal_sets(s1, s2).  
False
```

3 - Define o predicado *merge/3*, usado por exemplo em algoritmos de ordenação, tal que *merge(L1, L2, L3)* significa que *L3* é a lista ordenada formada pelos elementos de *L1* e de *L2*, as quais estão também ordenadas. *merge/3* não faz validações dos seus argumentos.

Exemplos

```
?- merge([1, 5, 7], [3, 10, 12], L).
```

```
L = [1, 3, 5, 7, 10, 12]
```

```
?- merge([], [5, 7, 9], L).
```

```
L = [5, 7, 9]
```