

1. (1 ponto) Use corte verde para fazer um predicado de obter o máximo entre dois números, de acordo com o exemplo a seguir:

```
?- max(3, 2, M).  
M = 3.
```

2. (1 ponto) Faça um predicado eficiente para verificar se um determinado conjunto de dígitos corresponde a um número ternário (ou seja, todos os dígitos são 0, 1 ou 2, de acordo com o exemplo a seguir:

Exemplo: ternario/2

```
?- ternario([0, 2, 1, 0]).  
true.
```

```
?- ternario([0, 2, 1, 3]).  
false.
```

3. (1 ponto) Usando `maplist`, faça um predicado que verifica se cada elemento de uma lista é simples, composto ou outra categoria e retorne essa informação em outra lista.

Exemplo: verificar/2

```
?- verificar([1, a(x), X, a], [simples, composto, outro, simples]).  
true.
```

```
?- verificar([1, a(x), X, a], R).  
R = [simples, composto, outro, simples].
```

4. (3 pontos) Considere a seguinte base de conhecimento em memória:

```
:- dynamic item/3.  
  
item(arroz, 10, alimento).  
item(feijao, 5, alimento).  
item(sabao, 8, limpeza).  
item(acucar, 7, alimento).  
item(carne, 16, alimento).  
item(tomate, 17, alimento).  
item(esponja, 3, limpeza).  
item(detergente, 4, limpeza).
```

4.1) Escreva um predicado que coloque em uma lista todos os produtos que forem do tipo alimento e que custem pelo menos 10.

4.2) Escreva um predicado que crie em memória fatos do tipo `alimento/2`, com o preço e nome de cada `item/3` correspondente em memória.

4.3) Escreva um predicado que escreva em disco, adicionando todos os fatos `item/3` descrevendo itens de limpeza a um arquivo chamado `limpeza.txt`.