Exercícios - Capítulos 5 e 6 Apostila Prolog.

5.1.

```
último([U], U).
último([_ | Resto], U) :-
    último(Resto, U).
```

5.2.

```
tam([], 0).
tam([_ | Resto], N) :-
    tam(Resto, NResto),
    N is NResto + 1.
```

5.3.

```
soma([], 0).
restantes.
soma([X | Resto], S) :-
    soma(Resto, SResto),
    S is X + SResto.
```

5.4.

```
máx([M], M).
máx([X | Resto], M) :-
    máx(Resto, MResto),
    (X > MResto -> M = X ; M = MResto).
```

5.5.

```
anexa([], L, L).
anexa([X | Resto], L, [X | Resultado]) :-
    anexa(Resto, L, Resultado).
inv([], []).
inv([X | Resto], Invertida) :-
    inv(Resto, RestoInvertido),
    anexa(RestoInvertido, [X], Invertida).
```

5.6.

```
anexa([], L, L).
anexa([X | Resto], L, [X | Resultado]) :-
    anexa(Resto, L, Resultado).
inv([], []).
inv([X | Resto], Invertida) :-
    inv(Resto, RestoInvertido),
    anexa(RestoInvertido, [X], Invertida).
sim(L) :-
    inv(L, Invertida),
    L = Invertida.
```

5.7.

```
d(0, zero).
d(1, um).
d(2, dois).
d(3, três).
d(4, quatro).
d(5, cinco).
d(6, seis).
d(7, sete).
d(8, oito).
d(9, nove).

txt([], []).
txt([Dígito | Resto], [Palavra | Resultado]) :-
    d(Dígito, Palavra),
    txt(Resto, Resultado).
```

5.8. a)

```
estrada(1,a,b).
estrada(2,a,d).
estrada(3,b,c).
estrada(4,c,d).
estrada(5,b,e).
estrada(6,c,f).
estrada(7,d,f).
estrada(8,e,f).
```

b)

```
rota(A, B, Rota) :-
    rota_aux(A, B, [A], Rota).
rota_aux(A, A, Acumulador, Rota) :-
    reverse(Acumulador, Rota).
rota_aux(A, B, Acumulador, Rota) :-
    estrada(_, A, CidadeAdjacente),
    \+ member(CidadeAdjacente, Acumulador),
    rota_aux(CidadeAdjacente, B, [CidadeAdjacente |
Acumulador], Rota).
```

5.9. a)

```
retângulo(A, B, C, D).

regular(retângulo(X1, Y1, X2, Y2)) :-
    X1 = X2,
    Y1 = Y2.
```

b)

```
quadrado(retângulo(X1, Y1, X2, Y2)) :-
    X1 = X2,
    Y1 = Y2,
    dif(X1, Y1).
```

6.1. Após a quarta consulta o resultado é:

```
metal(ouro).
metal(ferro).
metal(cobre).
metal(zinco).
```

"*retract(metal(X)).*" remove o primeiro fato metal(X) que unifica com a variável X. Neste caso, remove o fato metal(ouro).
Resultado após a quinta consulta:

```
metal(ferro).
metal(cobre).
metal(zinco).
```

6.2.

```
:- dynamic estado_lampada/1.

liga :-
    retractall(estado_lampada(_)),
    asserta(estado_lampada(acessa)).
desliga :-
    retractall(estado_lampada(_)),
    asserta(estado_lampada(apagada)).

lâmpada(Estado) :-
    estado_lampada(Estado).
```

6.3.

```
memorize(Fato) :-
    \+ call(Fato),  % Verifica se o fato já foi inserido
    asserta(Fato).

:- dynamic exemplo/1.

% Adicionar um fato
memorize(exemplo(a)).
memorize(exemplo(b)).
memorize(exemplo(a)).  % já existe
```

6.4.

```
% Base de dados
:- dynamic pos/2.

% Posição inicial
pos(robô, garagem).

% Exemplo de posição inicial dos objetos
pos(tv, sala).
pos(livro, quarto).
pos(prato, cozinha).

% Obter a posição de um objeto
pos(Obj, Loc) :-
    pos(Obj, Loc).

% Andar até uma determinada posição
ande(Dest) :-
    pos(robô, Origem),
    write('anda de '), write(Origem), write(' até '),
write(Dest), nl,
    retract(pos(robô, Origem)),
    asserta(pos(robô, Dest)).
```

```
% Pegar um objeto
pegue(Obj) :-
    pos(robô, Loc),
    pos(Obj, Loc),
    write('pega '), write(Obj), nl.

% Soltar um objeto
solte(Obj) :-
    pos(robô, Loc),
    write('solta '), write(Obj), nl.

% Exemplo
exemplo :-
    pos(O, L),
    write('O = '), write(O), write(', L = '), write(L), nl.
```

6.5.

```
% Base de dados
:- dynamic pos/2.

% Posição inicial
pos(robô, garagem).

% Exemplo de posição inicial dos objetos
pos(tv, quarto).
pos(livro, quarto).
pos(prato, cozinha).

% Obter a posição de um objeto
pos(Obj, Loc) :-
    pos(Obj, Loc).

% Andar até uma determinada posição
ande(Dest) :-
    pos(robô, Origem),
    write('anda de '), write(Origem), write(' até '),
write(Dest), nl,
    retract(pos(robô, Origem)),
    asserta(pos(robô, Dest)).
```

```
% Pegar um objeto
pegue(Obj) :-
    pos(robô, Loc),
    (pos(Obj, Loc) ->
        write('pega '), write(Obj), nl
    ;
        write('Onde está '), write(Obj), write('? '),
read(NovaLoc),
        asserta(pos(Obj, NovaLoc)),
        write('pega '), write(Obj), nl
    ).

% Soltar um objeto
solte(Obj) :-
    pos(robô, Loc),
    write('solta '), write(Obj), nl.

% Exemplo
exemplo :-
    pos(O, L),
    write('O = '), write(O), write(', L = '), write(L), nl.
```

6.6.

```
% Base de dados
:- dynamic pos/2.

% Posição inicial
pos(robô, garagem).

% Exemplo de posição inicial dos objetos
pos(tv, quarto).
pos(livro, quarto).
pos(prato, cozinha).

% Obter a posição de um objeto
pos(Obj, Loc) :-
    pos(Obj, Loc).

% Andar até uma determinada posição
ande(Dest) :-
    pos(robô, Origem),
    write('anda de '), write(Origem), write(' até '),
write(Dest), nl,
    retract(pos(robô, Origem)),
    asserta(pos(robô, Dest)).
```

```
% Pegar um objeto
pegue(Obj) :-
    pos(robô, Loc),
    (pos(Obj, Loc) ->
        write('pega '), write(Obj), nl
    ;
        write('Onde está '), write(Obj), write('? '),
read(NovaLoc),
        asserta(pos(Obj, NovaLoc)),
        write('pega '), write(Obj), nl
    ).

% Soltar um objeto
solte(Obj) :-
    pos(robô, Loc),
    write('solta '), write(Obj), nl.

% Levar um objeto até um local
leve(Obj, Dest) :-
    pos(robô, Origem),
    pos(Obj, Origem),
    ande(Dest),
    pegue(Obj),
    ande(Origem),
    solte(Obj).

% Exemplo
exemplo :-
    pos(O, L),
    write('O = '), write(O), write(', L = '), write(L), nl.
```