

BACKTRACKING EM PROLOG

```
p :- q, r, s.
```

```
q :- writeln('q1').  
q :- writeln('q2'),fail.
```

```
r :- writeln('r1').  
r :- writeln('r2').  
r :- writeln('r3'),fail.
```

```
s :- writeln('s1').  
s :- writeln('s2').  
s :- writeln('s3').  
s :- writeln('s4'),fail.
```

```
%%%%%%%%% Controlando Backtracking. %%%%%%%%%  
% num/2  
% num(Numero,Sinal)  
num(N,positivo) :- N>0.  
num(0,nulo).  
num(N,negativo) :- N<0.
```

```
% repeat/0  
% repeat :- repeat.  
% repeat.
```

```
% loop/0  
loop :- repeat,  
    write('Digite um número: '),  
    read(X), write(X),nl,  
    (X == 100),!.
```

PROCESSAMENTO DE LISTAS EM PROLOG

```
% Lista: é uma sequência ordenada de elementos que pode ter qualquer  
% comprimento. Os elementos da lista são separados por vírgula e colocados  
% entre um par de colchetes.
```

```
% Exemplos:
```

```
% 1. [a,b,c,d]
```

```
% 2. [1,4,7,2,10,3]
```

```
% 3. [] - lista vazia
```

```
% Uma lista é constituída de cabeça e cauda e denotada por:
```

```
% [Cabeça|Cauda]
```

```
% Cabeça: primeiro elemento da lista
```

```
% Cauda: o que restou com a exclusão da cabeça
```

```
% Exemplos:
```

```
% 1. [gosto,de,vinho] Cabeça: gosto Cauda: [de,vinho]
```

```
% 2. [[3],56,[2,7]] Cabeça: [3] Cauda: [5,[2,7]]
```

```
% 3. [X,Y|Z] Cabeça: X Cauda: [Y|Z]
```

```
% 4. [[o,gato]] Cabeça: [o,gato] Cauda: []
```

```
%
```

```
% UNIFICAÇÃO em Listas
```

```
% ?- [X|Y] = [a,b,c]
```

```
% X = a
```

```

% Y = [b,c]
%
% ?- Y = [b,c]
% Y = [b,c]
%
% ?- [X|Y] = [mesa]
% X = mesa
% Y = []
%
% ?- [X,Y|Z] = [a,b,c,d]
% X = a
% Y = b
% Z = [c,d]
%
% ?- [X,Y|Z] = [a]
% false
%
% ?- [[X,foi],[ao,cinema]] = [[ana,Y]|Z]
% X = ana
% Y = foi
% Z = [[ao,cinema]]
%
% ?- [X,Y,Z] = [ano,bissexta]
% false

```