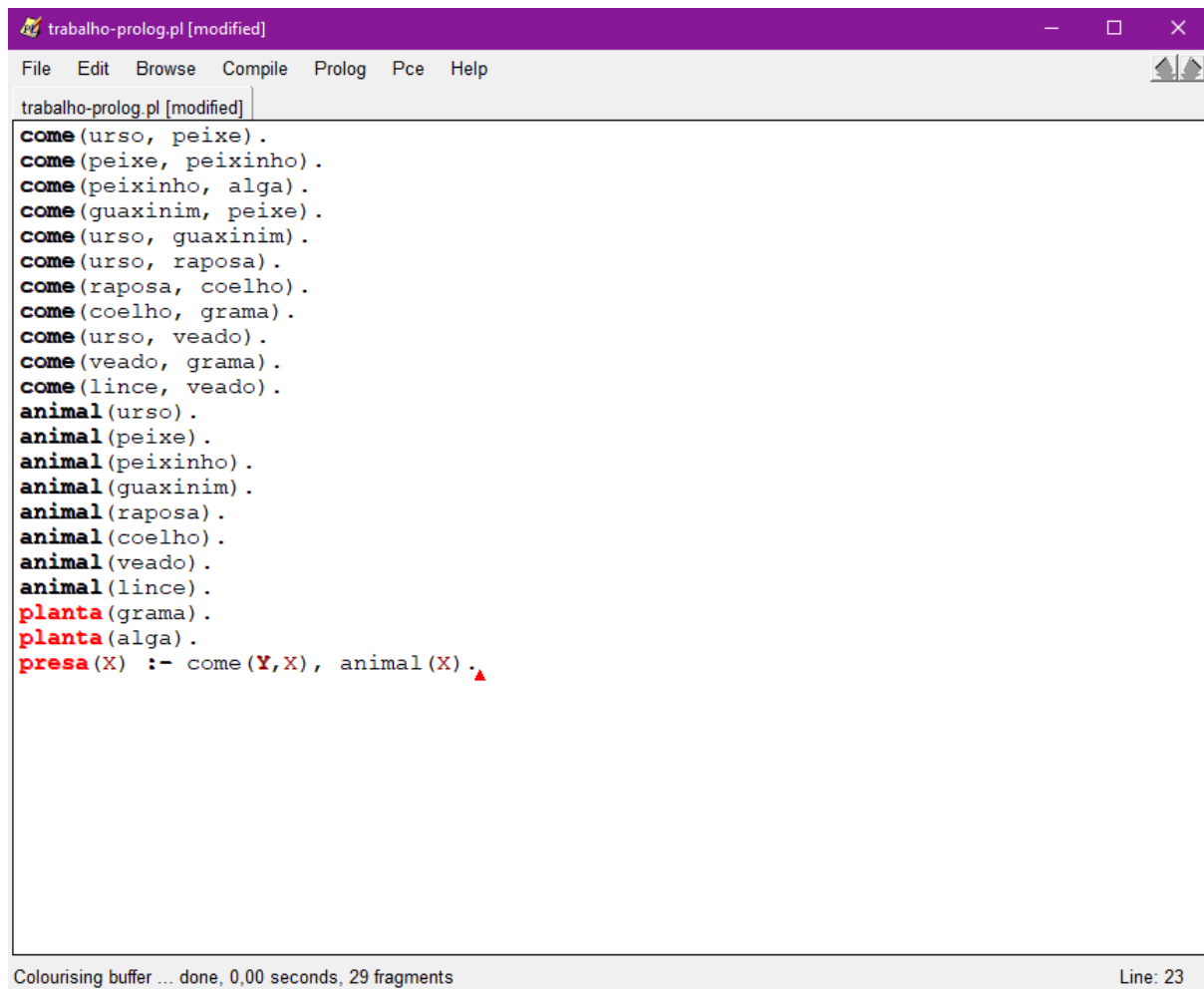


Banco de dados Prolog :



```
trabalho-prolog.pl [modified]
File Edit Browse Compile Prolog Pce Help
trabalho-prolog.pl [modified]
come(urso, peixe).
come(peixe, peixinho).
come(peixinho, alga).
come(guaxinim, peixe).
come(urso, guaxinim).
come(urso, raposa).
come(raposa, coelho).
come(coelho, grama).
come(urso, veado).
come(veado, grama).
come(lince, veado).
animal(urso).
animal(peixe).
animal(peixinho).
animal(guaxinim).
animal(raposa).
animal(coelho).
animal(veado).
animal(lince).
planta(grama).
planta(alga).
presa(X) :- come(Y,X), animal(X).
Colourising buffer ... done, 0,00 seconds, 29 fragments
Line: 23
```

A - Por que o Prolog é considerada uma definição recorrente ou recursiva?

R: O Prolog é considerada uma linguagem de programação recorrente ou recursiva porque permite que os programadores definam funções recursivas. A recursão é uma técnica de programação que permite que uma função se chame a si mesma. Isso pode ser usado para resolver problemas que seriam difíceis ou impossíveis de resolver sem recursão.

B- Explique porque são denominados como fatos (fato 1 e fato 3) os itens acima.

R: São denominados como fatos pois representam declarações verdadeiras sobre o mundo. Os fatos são declarações que não podem ser provadas e refutadas, apenas aceitas ou rejeitadas

C - Formule uma regra de Prolog que define o predicado *predador*.

```
trabalho-prolog.pl
File Edit Browse Compile Prolog Pce Help
trabalho-prolog.pl
come(urso, peixe).
come(peixe, peixinho).
come(peixinho, alga).
come(guaxinim, peixe).
come(urso, guaxinim).
come(urso, raposa).
come(raposa, coelho).
come(coelho, grama).
come(urso, veado).
come(veado, grama).
come(lince, veado).
animal(urso).
animal(peixe).
animal(peixinho).
animal(guaxinim).
animal(raposa).
animal(coelho).
animal(veado).
animal(lince).
planta(grama).
planta(alga).
presa(X) :- come(Y,X), animal(X).
predador(X) :- come(X, Y), animal(Y).▲

c:/users/usuario/documents/prolog/trabalho-prolog.pl compiled
Line: 23
```

D -Adicione essa regra ao banco de dados do Exemplo acima e diga qual seria a resposta à consulta.*?predador(X)*

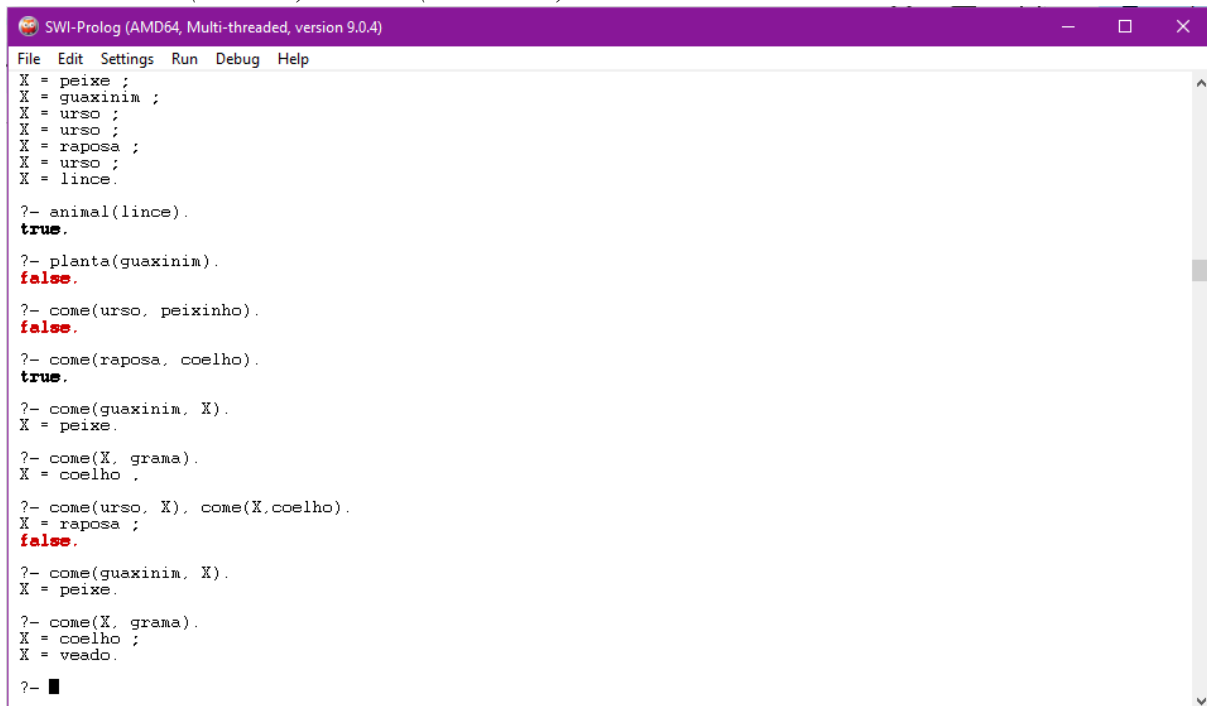
```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- predador(X).
ERROR: Unknown procedure: predador/1 (DWIM could not correct goal)
?-
Warning: c:/users/usuario/documents/prolog/trabalho-prolog.pl:22:
Warning: Singleton variables: [Y]
% c:/Users/Usuario/Documents/Prolog/Trabalho-Prolog.pl compiled 0.02 sec, 24 clauses
?- predador(X).
X = urso ;
X = peixe ;
X = guaxinim ;
X = urso ;
X = urso ;
X = raposa ;
X = urso ;
X = lince.
?-
```

E - Encontre os resultados da consulta em cada caso no Problema

?animal(lince)
?planta(guaxinim)
?come(urso, peixinho)
?come(raposa, coelho)
?come(guaxinim, X)
?come(X, grama)
?come(urso, X) e come(X, coelho)



```
File Edit Settings Run Debug Help
X = peixe ;
X = guaxinim ;
X = urso ;
X = urso ;
X = raposa ;
X = urso ;
X = lince.

?- animal(lince).
true.

?- planta(guaxinim).
false.

?- come(urso, peixinho).
false.

?- come(raposa, coelho).
true.

?- come(guaxinim, X).
X = peixe.

?- come(X, grama).
X = coelho.

?- come(urso, X), come(X, coelho).
X = raposa ;
false.

?- come(guaxinim, X).
X = peixe.

?- come(X, grama).
X = coelho ;
X = veado.

?-
```

F- Escreva, usando conceitos de Prolog, o significado de um **busca em profundidade**.

R: Em Prolog, uma busca em profundidade é um algoritmo que explora um grafo de acordo com uma ordem de profundidade. Isso significa que o algoritmo começa no nó raiz do grafo e explora todos os nós filhos do nó raiz antes de explorar qualquer outro nó.

G- Responda: Por que os conceitos de Prolog estão relacionados com a **lógica de predicados**? Faça uma sistematização com o conceito da regra de *Modus Ponens*

R: Os conceitos de Prolog estão relacionados com a lógica de predicados porque ambos são baseados no raciocínio dedutivo. A lógica de predicados é um sistema formal que permite a representação e manipulação de conhecimento sobre o mundo. O Prolog é uma linguagem de programação que implementa a lógica de predicados.

H- Por fim, faça uma conclusão do seu trabalho, procure na literatura (atenção para buscas científicas como, por exemplo, o *Google Scholar*) estudos sobre a lógica de predicados e a linguagem Prolog em que vocês entendam como interessantes no referido trabalho. Lembre que esse é o momento de finalizar o que foi desenvolvido pelos autores do trabalho.

R: Ao considerar estudos relevantes na literatura sobre lógica de predicados e PROLOG, é possível notar um vasto campo de aplicação para essas ferramentas. Pesquisas abordam desde sistemas de raciocínio automatizado e representação do conhecimento até a resolução de problemas complexos em domínios como inteligência artificial, processamento de linguagem natural e engenharia do conhecimento. A contínua evolução dessas áreas demonstra a relevância e o potencial dessas abordagens lógicas no mundo da computação e da pesquisa.