INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA MINAS GERAIS	**Ciência da Computação** Exercício em sala 03 Prolog.

Disciplina: Paradigmas de Linguagem	Data: ___/___/_____	
Professor: Bruno Ferreira	Turma: Integral	
Aluno	Valor: -	Nota: -

Objetivo: Reforçar os conceitos de programação lógica.

Forma de Entrega: Mostrar para o professor assim que terminar.

OBS: Será observada e avaliada a legibilidade do código sendo fundamental uma boa identação e a utilização de comentários.

---

1) Crie uma base de conhecimento com os fatos abaixo.

```
italiana(pizza).
italiana(calzone).
italiana(lazanha).
mineira(tutu).
mineira(feijoada).
gaucha(churrascada).
uruguaia(churrascada).
argentina(churrascada).
paraguaia(churrascada).
```

a) Crie uma regra com nome "brasileira" para dizer se o prato é típico brasileiro. Para ser um prato típico ele deve ser da culinária mineira ou gaúcha.
b) Consulte se lasanha é italiana
c) Consulte se tutu é comida brasileira
d) Consulte se existe comidas brasileiras
e) Consulte se há uma comida que seja brasileira e uruguaia.

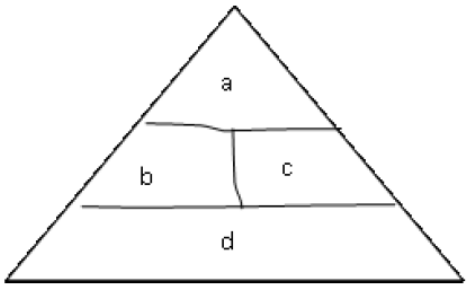
2) Crie uma base de conhecimento com os seguintes quaternários. Os fatos representam dados dos alunos, disciplina, nota e frequência.

```
nota_freq(alberto, logica, 9.0, 60.0).
nota_freq(juca, logica, 7.0, 95.0).
nota_freq(maria, logica, 4.0, 75.0).
```

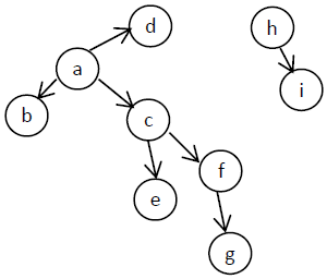
a) Faça uma consulta à base de conhecimento para retorna apenas o nome e a nota dos alunos.
Dica: O *underscore (variável anônima)* faz com que o PROLOG ignore os argumentos.
b) Crie um regra com a conclusão no formato "aprovado(X, Y, Z, W)". Ela é responsável por informar se o aluno está aprovado considerando que a nota seja maior ou igual a 7 e a frequência maior ou igual a 75.
c) Consulte pelos aprovados em qualquer disciplina.
d) Consulte pelos aprovados em lógica
e) Incluindo uma regra que permita determinar os alunos reprovados em qualquer disciplina.
f) Consulte pelos reprovados em qualquer disciplina

3) Faça um código em PROLOG que tem uma base de conhecimentos representando o estado atual de um tabuleiro do jogo da velha e diga se há vencedor (cruz ou bola).

4) Faça um código em PROLOG que dado um mapa com quatro regiões sendo que somente 3 delas são adjacentes. O programa deve escolher, entre três opções, a cor para cada região de forma que duas regiões fronteiriças não tenham a mesma cor.



5) Represente em uma base de conhecimento um mapa de estradas que interconectam cidades dado pelas árvores abaixo:



a) Duas cidades estão conectadas se há uma ligação direta entre elas ou se há uma ligação indireta entre elas. Assim, crie as regras responsáveis por essa consulta.
b) Dada uma cidade, deseja-se saber todas as cidades alcançáveis a partir dela.
c) Dadas duas cidades deseja-se saber se há conexão ou não entre elas.

6) Insira a seguinte base de conhecimento no Prolog e responda as questões a seguir.

```
come(urso,peixe).
come(peixe,peixinho).
come(peixinho,alga).
come(peixe,alga).
come(urso,raposa).
come(veado,grama).
come(peixe,minhoca).
come(urso,guaxinim).
come(raposa,coelho).
come(urso,veado).
come(lince,veado).
come(planta_carnívora,mosca).
come(veado,planta_carnívora).
animal(urso).
animal(peixe).
animal(raposa).
animal(veado).
animal(minhoca).
animal(lince).
animal(coelho).
animal(guaxinim).
```

```
animal(mosca).
animal(peixinho).
planta(grama).
planta(alga).
planta(planta_carnívora).
```

a) Quais são os elementos que comem peixe?
b) Quais são os elementos com propriedade animal?
c) Quais são todos os elementos que participam da relação come?
d) Escreva as seguintes regras:
   carnívoro(X) = quem como animal
   herbívoro(X) = come planta e não come animal
   predador(X) = é carnívoro e também é animal
   presa(X) = é quem é comido por predador e também é animal
   caçado(X) = caçado é quem é presa
   pertence_a_cadeia(X,Y) =X pertence a cadeia alimentar de Y
e) Faça as seguintes consultas
   Peixe come peixinho e minhoca?
   Quais são as plantas?
   Quem é comido pelo urso?
   Quem come peixe?
   Quem é predador?
   Quem é predador e também presa?
   Quem é presa e herbívoro?
   Quem pertence a cadeia alimentar do urso?
   Quem pertence a cadeia alimentar do urso e ao mesmo tempo come planta?
   A minhoca pertence a cadeia alimentar de quem?

7) Implementar um programa que determina se um determinado dia faz parte de um dia da semana ou final de semana. Note que a determinação de uma categoria já exclui a possibilidade de que o elemento pertença a outra categoria, ou seja, não existe um dia que seja semana e final de semana. Assim, você pode usar o comando de corte.

8) Insira a seguinte base de conhecimento no Prolog e responda as questões a seguir.

```
homem(pedro).
homem(marcos).
homem(ze).
mulher(maria).
mulher(joana).
idade(ze,30).
idade(maria,40).
idade(marcos,20).
idade(pedro,25).
idade(joana,28).
gosta(ze,aventura).
gosta(maria,comedia).
gosta(joana,romance).
gosta(marcos,terror).
gosta(marcos,romance).
gosta(pedro,romance).
gosta(maria,romance).
opcao(ze,20,40).
opcao(pedro,25,55).
opcao(marcos,20,21).
opcao(maria,25,55).
opcao(joana,28,30). // significa que a joana gostaria de se relacionar com pessoas etnre 28 a 30 anos
```

Cria as seguintes regras:

a) afinidade_filme(X,Y) Seleciona pessoas com o mesmo gosto de filme (sendo X e Y pessoas, X possui o mesmo gosto de filme do que Y).

b) casal(X,Y) Seleciona casais (sexos diferentes)
c) casal_afinidade_filme(X,Y)Seleciona casais com o mesmo gosto de filme
d) casal_afinidade_idade(X,Y) Seleciona casais cujos gostos de faixa etárias sejam atendidos
e) casal_total(X,Y)Seleciona casais com afinidade em filme e faixa etária
f) Faça consulta diversas para cada regra.

9) Imagine um mapa em um quadriculado de 4 x 4 posições, com cada posição tendo coordenadas (x, y). Crie os seguintes predicados:

- existe_acima
- existe_abaixo
- existe_direita
- existe_esquerda

Cada predicado tem 2 parâmetros: uma coordenada x e uma coordenada y. Os predicados devem ter valor verdadeiro se existe uma posição no mapa na posição indicada (acima, abaixo, à direita ou à esquerda). Por exemplo, se a posição(1, 1) fica no canto inferior esquerdo do mapa, não existe nenhuma posição abaixo dela, mas existe uma posição acima. Daí os seguintes exemplos de consulta:

```
?- existe abaixo(1, 1).
false .
?- existe acima(1, 1).
true.
```

10) Com a mesma idéia de mapa 4x4 da questão anterior, crie os predicados:

- pos_acima
- pos_abaixo
- pos_direita
- pos_esquerda

Cada predicado deve ter 4 parâmetros, sendo os 2 primeiros as coordenadas (x; y) da posição inicial, e os 2 últimos parâmetros sendo as coordenadas (x; y) da posição acima, abaixo, à direita ou à esquerda da posição inicial, se existir. Se não existir a posição indicada, as 2 ultimas coordenadas devem ser iguais às coordenadas iniciais.
Por exemplo, para a posição (1; 1):

```
?- pos_acima(1, 1, X, Y).
X = 1, Y = 1.
?- pos_abaixo(1, 1, X, Y).
X = 1, Y = 2.
```

11) Dados os seguintes fatos que listam os últimos presidentes do Brasil:

```
presidente(sarney, 1985, 1990).
presidente( collor , 1990, 1992).
presidente(itamar, 1992, 1995).
presidente(fhc, 1995, 2003).
presidente( lula , 2003, 2011).
presidente(dilma,2012,2018).
```

a) Crie uma regra presidente_em(A, X) representando o fato que X era presidente no ano A. Por exemplo, para os fatos acima, a seguinte consulta deve ser possível:

```
?- presidente em(1991, X).
X = collor ;
false .
```

12) A base de dados abaixo representa uma tabela que relaciona a cada país sua área em Km^2 e sua população, ambos em escala de milhões.

```
% país(Nome, Área, População)
país(brasil, 9, 130).
país(china, 12, 180).
país(eua, 9, 230).
país(índia, 3, 450).
```

a) Com base nessas informações, determine a densidade demográfica do Brasil. A densidade é a divisão da população pela área.
b) Qual a diferença entre a população da China e da Índia?
c) A área do Brasil é igual à área dos Estados Unidos?
d) A população dos Estados Unidos é maior que a população da Índia?

Exercício com Listas

1) Crie uma regra que retorne os dois primeiros elementos de uma lista.

2) Crie uma regra que retorne "true" se duas listas tem a mesma quantidade de elementos.

3) Crie uma regra que retorne o último elemento de uma lista.

4) Crie uma regra para ver se um elemento pertence a lista

5) Verifique se dois elementos são consecutivos. Dois elementos E1 e E2 são consecutivos se eles forem o primeiro e o segundo elementos da lista ou, se forem consecutivos na cauda da lista.

6) Crie uma regra para saber o tamanho (qtde de elementos) de uma lista

7) Crie uma regra para somar os elementos de uma lista

8) Crie uma regra que retorne o e-nésimo elemento de uma lista. Lembre-se o primeiro elemento de uma lista é a cabeça da lista e o e-nésimo elemento de uma lista é o (n-1)-ésimo elemento da cauda.

9) Pegar elementos de uma lista dada a lista de posições. pegar(Lista de posições, lista a ser pesquisada, lista resultante). Lembre-se, Pegar nenhum elemento de uma lista é obter uma lista vazia e seja a lista de posições [M|N] e a lista de elementos L; pegar os elementos de L especificados na lista de posições significa pegar o M-ésimo elemento de L e os elementos especificados na cauda N da lista de posições.

10) Crie uma regra que receba uma lista e retorne uma lista com os elementos duplicados. Atenção, se a lista for a lista vazia, o resultado é a lista vazia, caso contrário, duplicar os elementos de uma lista cuja a cabeça é X e a cauda é Tail, significa duplicar a cabeça, e duplicar os elementos de Tail

11) Crie uma regra que insira um elemento na primeira posição da lista. Inserir um elemento na lista, significa criar uma nova lista na qual, o novo elemento é o cabeça.

12) Concatenar a primeira lista com a segunda. Se a primeira lista for uma lista vazia, então a segunda lista é a lista resultante. Se a primeira lista não for vazia, retira um elemento e coloque na lista resultante. Veja uma possível chamada: concatena(Lista1,Lista2,ListaResultante).

13) Chame a função concatena (do exercício anterior) passando duas variáveis aos dois primeiros parâmetros e uma lista no terceiro parâmetro. O que aconteceu?

Fim