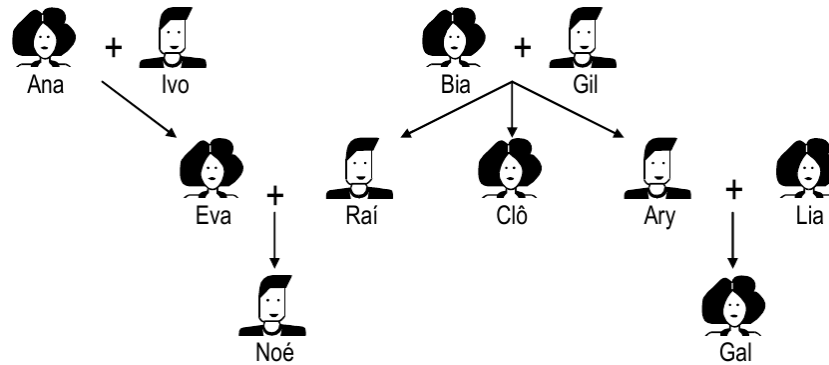


Lista de Exercícios 7 – Paradigma Lógico (Prolog) (Incompleta)

1) Considere a árvore genealógica a seguir:



a) Usando fatos, defina as relações:

pai(X,Y) ↔ “X é pai de Y.”

mae(X,Y) ↔ “X é mãe de Y”

homem(X) ↔ “X é homem”

mulher(X) ↔ “X é mulher”

b) Usando regras, defina a relação **gerou(X,Y)** que denota “X gerou Y se X é pai ou mãe de Y”.

c) Usando regras, defina as relações **filho(X,Y)**, **filha(X,Y)**, **tio(X,Y)**, **tia(X,Y)**, **primo(X,Y)**, **prima(X,Y)**, **avô(X,Y)** e **avó(X,Y)**.

2) Suponha uma base de dados com fatos do tipo **filho(X,Y)** que denota que X é filho de Y. Codifique regras equivalentes às seguintes sentenças:

a) *Todo mundo que tem filhos é feliz.*

b) *Um casal é formado por duas pessoas que têm filhos em comum.*

3) Seja o seguinte código Prolog:

```
%país(Nome, Área, População)
país(brasil, 9, 130).
país(china, 12, 1800).
país(eua, 9, 230).
país(índia, 3, 450).
```

Inclua uma regra para o predicado **dens(P,D)**, que relaciona cada país P à sua densidade demográfica correspondente D. Em seguida, faça consultas para descobrir:

a) *Qual a densidade demográfica de cada um dos países.*

b) *Se a Índia é mais populosa que a China.*

4) Seja o seguinte código Prolog:

```
% filme(Título, Gênero, Ano, Duração)
filme('Uma linda mulher', romance, 1990, 119).
filme('Sexto sentido', suspense, 2001, 108).
filme('A cor púrpura', drama, 1985, 152).
filme('Copacabana', comédia, 2001, 92).
filme('E o vento levou', drama, 1939, 233).
filme('Carrington', romance, 1995, 130).
```

Inclua as seguintes regras:

a) *Um filme é um clássico se ele foi lançado até 1985.*

b) *Um filme é longo se tem duração superior a 150 minutos.*

5) O quadro a seguir relaciona a cada funcionário de sua empresa seu *código*, seu *salário* e os seus *dependentes*. Faça o que cada item pede:

Código	Nome	Salário	Dependentes
1	André	R\$ 1000	Carla
2	Bruno	R\$ 1200	----
3	Carlos	R\$ 903	Rute e Diana
4	Diego	R\$ 2500	Maria e Daniel
5	Eduardo	R\$ 1800	Paulo
6	Flavio	R\$ 1100	----

- Escreva fatos do tipo **func(Código, Nome, Salário)** que relacionem estas informações de cada funcionário da empresa.
- Escreva fatos do tipo **dep(Código, Nome)** que relaciona o código de um funcionário com o nome de um de seus dependentes.
- Escreva a consulta: *Quem são os dependentes de Carlos?*
- Escreva a consulta: *Quem tem salário entre R\$ 1.500 e R\$ 3.000?*
- Escreva a consulta: *Quem não tem dependentes e ganha menos de R\$ 1.200?*
- Escreva a consulta: *Quem depende de funcionário que ganha mais de R\$ 1.700?*

6) Codifique um programa contendo as informações da tabela abaixo e faça as consultas indicadas a seguir:

Nome	Sexo	Idade	Altura	Peso
Ana	fem	23	1,55	56,0
Bia	fem	19	1,71	61,3
Luke	masc	22	1,80	70,5
Lia	fem	17	1,85	57,3
Eva	fem	28	1,75	68,7
Rob	masc	25	1,72	68,9

- Quais as mulheres com mais de 20 anos de idade?
- Quem tem pelo menos 1,70m de altura e menos de 65kg?
- Quais são os possíveis casais onde o homem é mais alto do que a mulher?

7) O programa a seguir associa a cada pessoa seu esporte preferido.

```
joga(ana,volei).
joga(bia,tenis).
joga(cid,basquete).
joga(eva,volei).
joga(leo,tenis).
```

Suponha que desejamos consultar esse programa para encontrar um parceiro **P** para jogar com **Leo**. Então, podemos realizar essa consulta de duas formas:

- ?- **joga(P,X), joga(leo,X), P \= leo.**
- ?- **joga(leo,X), joga(P,X), P \= leo.**

Desenhe as árvores de busca construídas pelo sistema ao responder cada uma dessas consultas. Qual consulta é mais eficiente?

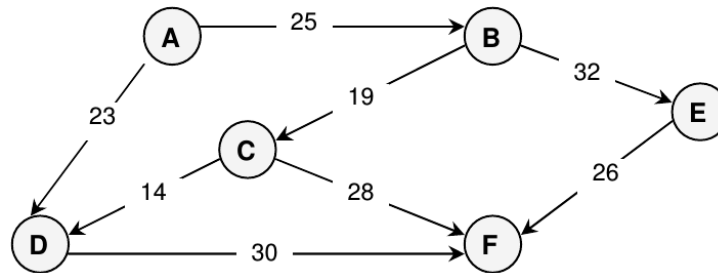
8) Defina um procedimento recursivo para calcular o produto de dois números naturais usando apenas somas e subtração.

9) Defina os seguintes predicados sobre listas:

- a) **ultimo(L,U)** \leftrightarrow Determina o último elemento *U* numa lista *L*.
- b) **tam(L,N)** \leftrightarrow Determina o número *N* de elementos numa lista *L*.
- c) **soma(L,S)** \leftrightarrow Determina a soma *S* dos itens da lista *L*.
- d) **máx(L,M)** \leftrightarrow Determina o maior item *M* na lista *L*.

10) Usando fatos do tipo **d(0,zero)**, **d(1,um)**, ..., **d(9,nove)**, defina o predicado **txt(D,P)** que converte uma lista de dígitos numa lista de palavras. Por exemplo, **txt([7,2,1],P)** resulta em **P = [sete,dois,um]**.

11) O grafo a seguir representa um mapa, cujas cidades são representadas por letras e cujas estradas (de sentido único) são representadas por números, que indicam sua extensão em km.



- a) Usando o predicado **estrada(Origem, Destino, Km)**, crie um programa para representar esse mapa.
- b) Defina a relação transitiva **dist(A,B,D)**, que determina a distância *D* entre duas cidades *A* e *B*.
- c) Defina o predicado **rota(A,B,R)**, que determina uma rota *R* (lista de cidades) que leva a cidade *A* até a cidade *B*.

12) Considere a seguinte base de fatos Prolog:

```
%carro(Dono, NomeCarro, dataCompra(Dia, Mes, Ano), Quilometragem, placa(Cidade, Letras, Numeros))
carro(carlos, silverado, dataCompra(15, 08, 2001), 46800, placa(russas, 'LHR', 1178)).
carro(jose, corsa, dataCompra(06, 11, 2004), 23400, placa(fortaleza, 'AJV', 2273)).
carro(maria, clio, dataCompra(19, 12, 2004), 25730, placa(russas, 'CKP', 5194)).
```

Dada a consulta a seguir, escreva o significado dessa consulta em português:

?- carro(X,_,dataCompra(_,_,2001),_,placa(russas,_,_)).

13) Considere a seguinte base de fatos Prolog:

```
%autor(Nome, PaisDeNascimento).
autor(stephenKing, eua).
autor(georgeMartin, eua).
autor(jkRowling, reinoUnido).

%livro(Nome, NumeroPaginas, Genero, NomeAutor).
livro(cemiterio_maldito, 374, terror, stephenKing).
livro(aGuerraDosTronos, 694, fantasia, georgeMartin).
livro(morteSubita, 512, romance, jkRowling).
```

Dada a consulta a seguir, escreva o significado dessa consulta em português.

?- livro(X,_,_,Y), autor(Y,reinoUnido).

14) Defina uma regra em Prolog que remova o último elemento de uma lista. Por exemplo:

```
?- removeUltimo([a,b,c,d],X).
X = [a,b,c]
```

15) Seja a base de fatos Prolog abaixo:

```
%governou(X,I,F) ⇔ O presidente X governou o Brasil do ano I ao ano F.
governou(fernandoCollor ,1990,1992).
governou(itamarFranco ,1992,1995).
governou(fernandoHenrique,1995,2003).
governou(lula ,2003,2011).
```

Escreva a regra **sucessor(X,Y)** que indique que o presidente Y assumiu imediatamente após o presidente X.

16) Considere a seguinte base de fatos Prolog:

```
%animal(Especie,Nome).
animal(dragao,balerion).
animal(dragao,drogon ).
animal(lobo ,fantasma).
animal(cavalo,prata ).

%castelo(Nome).
castelo(winterfell ).
castelo(rochedoCasterly).

%vivo(X) ⇔ X está certamente vivo.
vivo(jonSnow ).
vivo(daenerys).
vivo(drogon ).

%morto(X) ⇔ X está certamente morto.
morto(aegon ).
morto(nedStark).
morto(balerion).
morto(prata ).

%possui(X,Y) ⇔ Pessoa X possui Y.
possui(aegon ,balerion ).
possui(jonSnow ,fantasma ).
possui(daenerys ,drogon ).
possui(daenerys ,prata ).
possui(nedStark ,winterfell ).
possui(tywinLannister,rochedoCasterly).
```

Escreva em Prolog:

- uma regra **guardiao(X)** que diga que quem possui castelo é guardião de um dos sete reinos.
- uma consulta para saber que pessoas têm dragões.
- uma consulta para saber que pessoas não sabem se seu animal está vivo ou morto.

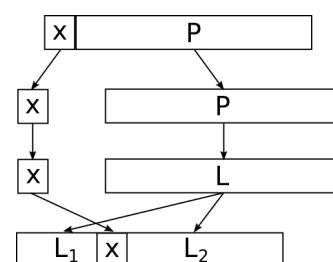
17) No jogo PokemonGO temos um conjunto de seres denominados pokemon. Cada pokemon possui um tipo específico. Alguns tipos de pokemon possuem vantagens sobre outros. Quando o tipo de um pokemon possui vantagem sobre o tipo do outro, seu ataque tira o dobro de dano que ele causaria normalmente. Considere a seguinte base de fatos Prolog:

<pre>%pokemon(Nome,Tipo,Dano). pokemon(bulbassauo,planta ,100). pokemon(caterpie ,inseto ,60). pokemon(pikachu ,eletrico,120). pokemon(charmander ,fogo ,110). pokemon(squirtle ,agua ,110). pokemon(pidgeotto ,voador ,90). pokemon(onix ,pedra ,100). pokemon(rhydon ,terra ,150).</pre>	<pre>%vantagem(Tipo1,Tipo2). %Pokemon Tipo1 possui vantagem sobre pokemon Tipo2 vantagem(eletrico,agua). vantagem(terra ,eletrico). vantagem(planta ,pedra). vantagem(planta ,agua). vantagem(fogo ,planta). vantagem(fogo ,inseto). vantagem(agua ,fogo).</pre>
--	--

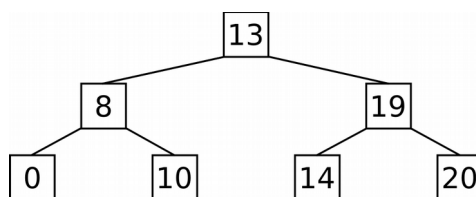
Faça o que se pede:

- Construa a consulta “Que tipos são fracos contra o tipo fogo?”
- Construa a consulta “Quais os pokemons que possuem desvantagem com relação ao pikachu?”
- Construa a regra **dano(A,B,D)** que calcule o dano D que um pokemon A causa no pokemon B.

18) (InsertionSort é um algoritmo de ordenação que, dada uma lista com n números, ele vai inserindo os elementos um a um em uma lista, inicialmente vazia, colocando os elementos na ordem correta. A figura ao lado mostra um esboço da ideia: o elemento x é reservado, o restante da lista é ordenado recursivamente, e x é inserido na posição apropriada nesta nova lista ordenada. Implemente esse algoritmo utilizando Prolog.



- 19) Uma árvore binária pode ser representada em Prolog através de dois fatos:
- o fato **node(A,B,C)** indica que o nó **A** é pai dos nós **B** (à esquerda) e **C** (à direita)
 - o fato **leaf(A)** indica que **A** é uma folha na árvore.



- a) Escreva um conjunto de fatos representando a árvore mostrada acima.
b) Escreva uma regra **busca(X,N)** que indique quando o elemento **X** está na árvore com raiz **N**.

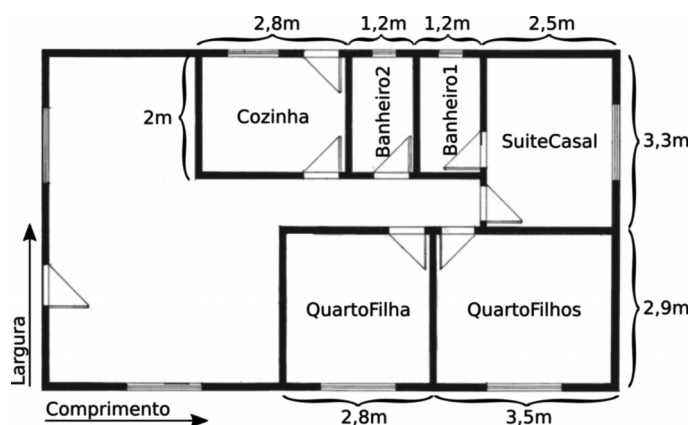
20) Escreva em Prolog uma regra **rem(X,L1,L2)** em que **L2** é igual a lista **L1** com o elemento **X** removido caso ele esteja presente em **L1**. Dois exemplos de consultas usando a regra **rem**:

```

?- rem(c,[a,b,c,d,e],L).
L = [a,b,d,e]
?- rem(z,[a,b,c,d,e],L).
L = [a,b,c,d,e]
  
```

21) A planta baixa ao lado pertence a casa de uma família. Escreva em Prolog:

- a) Fatos indicando o comprimento e a largura de cada cômodo nomeado na figura.
b) Uma regra **area(X,A)** que denota que **A** é a área do cômodo **X**.
c) Uma consulta perguntando se o quarto da filha é maior, em área, do que a suíte do casal.



22) Os triângulos podem ser classificados segundo o comprimento dos seus lados. Se um triângulo possui todos os lados iguais, ele é dito equilátero. Se um triângulo possui pelo menos dois lados iguais, ele é dito isósceles. Se um triângulo possui os 3 lados de diferentes comprimentos, ele é dito escaleno. Seja uma base de fatos prolog **tri(T,A,B,C)** que denota que um triângulo **T** possui lados de comprimento **A**, **B** e **C**. Escreva regras **eq(T)**, **iso(T)** e **esc(T)** que indiquem quando um triângulo **T** é equilátero, isósceles e escaleno, respectivamente.

23) Seja uma base de dados com fatos da forma:

```

%C = código do funcionário
%N = nome do funcionário
%D = código do departamento em que o funcionário trabalha
%S = salário do funcionário
func(C,N,D,S).

...
%D = código do departamento
%N = nome do departamento
%L = local do departamento
%C = código do funcionário que gerencia o departamento
dept(D,N,L,C).
  
```

Defina uma regra **chefe(N,M)** que denota que **N** é o nome do chefe do funcionário de nome **M**. Obs: O gerente do departamento é chefe de todos os funcionários daquele departamento.

24) Considere a seguinte base de fatos Prolog:

```
%carro(Codigo,Modelo,Ano,Quilometragem,placa(Cidade,Letras,Numeros)).
carro(3371,gol,2000,150000,placa(russas,ahr,1178).
carro(3411,corssa,2005,56000,placa(fortaleza,ajv,2273).
carro(3382,clio,2012,12000,placa(russas,ckp,5194).

%motorista(Nome,CodigoCarro,dataCompra(Dia,Mês,Ano)).
motorista(carlos,3371,dataCompra(15,08,2010).
motorista(jose,3411,dataCompra(06,11,2016).
motorista(maria,3382,dataCompra(19,12,2012).
```

Faça uma consulta em Prolog para descobrir os motoristas de carros com placa de Russas, com carros com menos de 20000 Km rodados.

25) Estudiosos já formularam diversas séries matemáticas que, se levadas ao infinito, podem calcular π de forma precisa em inúmeras casas decimais. Algumas delas são tão complexas que só conseguem ser analisadas por supercomputadores. A de Gregory-Leibniz, por sua vez, é uma das mais simples. Para isso, usa-se a seguinte fórmula:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} - \frac{4}{15} \dots$$

Somas e subtrações são alternadas com frações de numerador 4 e denominador com números ímpares em sequência. Quanto mais longe for, mais perto este resultado será de π . Escreva um predicado em Prolog **pi(N,E)**, onde **N** é o número de parcelas da série e **E** é a estimativa de π .