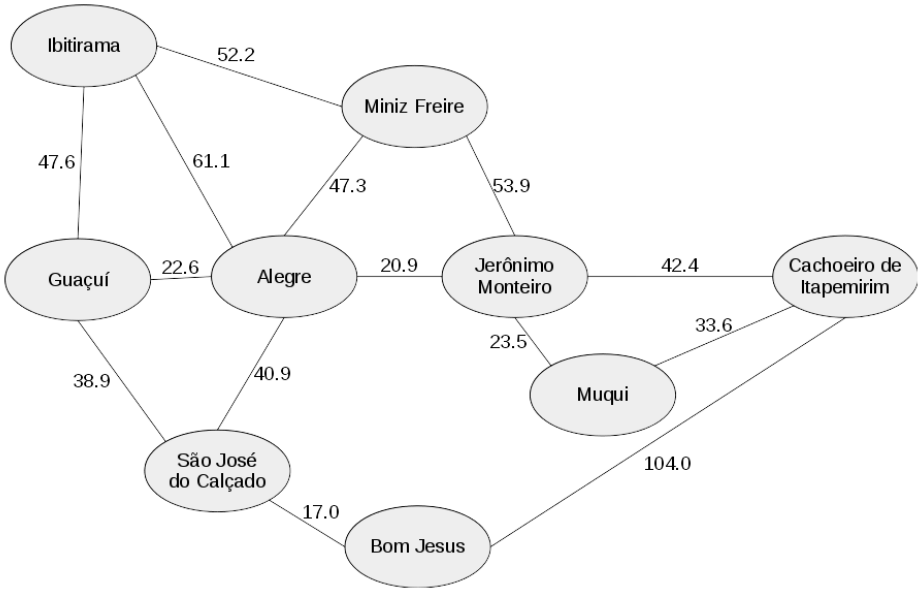
Exercícios – Prolog

1. Verificar se o elemento Elem existe na lista. Ex:
```
membro(Lista, Elem).
?- membro([3,1,7,2,0], 7).
true.
```
2. Encontrar o elemento da posição N de uma lista. Ex:
```
itemDaLista(Lista, Elem, N).
?- itemDaLista([3,1,7,2,0], Elem, 3).
Elem = 7.
```
3. Contar quantos elementos Elem existem na lista. Ex:
```
contarElemDaLista(Lista, Elem, Qnt).
?- contarElemDaLista([3,2,3,1,3,8,3], 3, Qnt).
Qnt = 4.
```
4. Encontrar números consecutivos em uma lista. Ex:
```
consecutivosNaLista(Lista, Elem).
?- consecutivosNaLista([1,2,3,3,4], 3).
true.
```
5. Encontrar se existe um elemento repetido na lista. Ex:
```
elementoRepetidoNaLista(Lista).
?- consecutivosNaLista([1,2,3,4,3]).
true.
```
6. Contar o tamanho da lista. Ex:
```
tamLista(Lista, Tam).
?- tamLista([3,5,8,2,4,9], Tam).
Tam = 6.
```
7. Remover um item da lista:
```
removerItemDaLista(Lista, Item, NovaLista).
?- removerItemDaLista([3,5,8,2,4,9], 4, NovaLista).
NovaLista = [3,5,8,2,9].
```
8. Fornecer o último elemento de uma lista.
```
ultimoElemDaLista(Lista, Elem).
?- ultimoElemDaLista([3,2,3,1,3,8], Elem).
Elem = 8.
```
9. Encontrar o maior elemento da lista.
```
maiorElemDaLista(Lista, Elem).
?- maiorElemDaLista([3,2,3,1,3,8], Elem).
Elem = 8.
```
10. Encontrar o menor elemento da lista.
```
menorElemDaLista(Lista, Elem).
?- menorElemDaLista([3,2,3,1,3,8], Elem).
Elem = 1.
```
11. Somar todos os número da lista.
```
somarLista(Lista, Soma).
?- somarLista([3,2,3,1,3,8], Soma).
Soma = 20.
```
12. Encontrar o índice de um número (da esquerda para a direita). Ex:
```
indiceDe(Num, Lista, Idx).
?- indiceDe(7, [4,7,2,3,5,8], Idx).
Idx = 2.
```
13. Encontrar o índice de um número (da direita para a esquerda). Ex:
```
indiceDe(Num, Lista, Idx).
?- indiceDe(7, [4,7,2,3,5,8], Idx).
Idx = 5.
```
14. Encontrar qual é o índice que tem o menor número da lista.
```
indiceMenorDaLista(Lista, Idx).
?- indiceMenorDaLista([4,7,2,1,5,8], Idx).
Idx = 4.
```
15. Encontrar qual é o índice que tem o maior número da lista.
```
indiceMaiorDaLista(Lista, Idx).
```

```
?- indiceMaiorDaLista([4,7,2,1,5,8], Idx).
Idx = 6.
```

16. Inserir um item em uma lista já ordenada. Ex:
```
inserirNaListaOrdenada(Item, Lista, ListaOrd).
?- InserirNaListaOrdenada(4, [3,6,7,8,10], ListaOrd).
ListaOrd = [3,4,6,7,8,10].
```
17. Ordenar uma lista. Ex:
```
ordenarLista(Lista, ListaOrd).
?- ordenarLista([2,1,3,4,2,5,6], ListaOrd).
ListaOrd = [1,2,2,3,4,5,6].
```
18. Somar um inteiro a todos os itens de uma lista. Ex:
```
somarIntNaLista(Lista, Numero, ListaSomada).
?- somarIntNaLista([1,2,3,4,5], 2, ListaSomada).
ListaSomada = [3,4,5,6,7].
```
19. Cria uma lista com todos os filhos apresentados no exercício 1. Ex.:
```
listaDeFilhos(Pai, Lista).
?- listaDeFilhos(alfredo, L).
L = [suse, joana].
```
20. Criar a sequência de Fibonacci com *N* números. Ex:
```
fibonacci(N, Resultado).
?- fibonacci(5, R).
R = [1, 1, 2, 3, 5].
```
21. Calcular o fatorial de *N*. Ex:
```
fatorial(N, Resultado).
?- fatorial(3, R).
R = 6.
```
22. Somar os números pares de uma lista. Ex:
```
somarPares(Lista, Soma).
?- somarPares([1,2,3,4,5,6], R).
R = 12.
```
23. Somar os números ímpares de uma lista. Ex:
```
somarImpares(Lista, Soma).
?- somarImpares([1,2,3,4,5,6], R).
R = 9.
```
24. Fazer a média de uma lista. Ex:
```
media(Lista, Resultado).
?- media([1,2,3,4,5], R).
R = 3.
```
25. Verificar quantas vezes o elemento se repete na lista. Ex:
```
repete(Lista, Elem, Resposta).
?- repete([a,b,a,d,b,c,a,d,b,e,d], a, R).
R = 3.
```
26. Buscar todos os elementos diferentes de um item na lista.. Ex:
```
diferenteDe(Lista, Elem, Resposta).
?- diferenteDe([a,b,c,d,b,e,d,a,1,b], b, R).
R = [a,c,d,e,d,a,1].
```
27. Criar uma árvore genealógica com sua família (pais, avôs, irmãos, tios)
    Crie predicados capazes de responder:
    - quem são os pais de alguém
    - quem são os tios de alguém
    - quem são os avôs de alguém
    - quem são os netos de alguém
    - quem são os sobrinhos de alguém.
28. Com um dominó possuindo somente as peças: 0-0, 0-1, 1-1, 0-2, 1-2, 2-2, 0-3, 1-3, 2-3, 3-3.
    Faça:
    - A representação dessas peças no Prolog.
    - O sorteio de peças entre você e seu oponente.
    - A escolha da melhor peça a jogar em um instante do jogo.

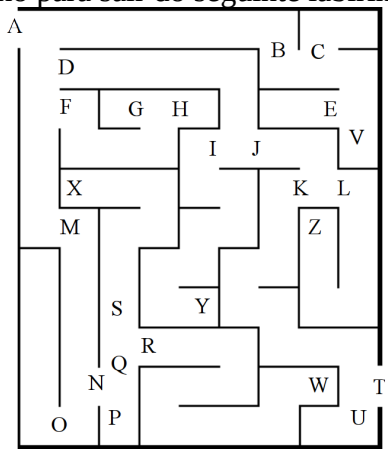
29. Com o seguinte mapa:



Faça os predicados que sejam capazes de responder:

- Quais são as cidades adjacentes a cidade X.
- Quais são os caminhos entre a cidade X e a Y.
- Qual é o menor caminho entre a cidade X e a Y.
- Qual é o menos caminho entre a cidade X e a Y, mas sem passar pela cidade Z.

30. Usando busca, encontre o caminho para sair do seguinte labirinto:



31. Faça um predicado e as regras capazes de armazenar as seguintes informações:

- José, nota 7
- Alaor, nota 4.6
- Nilza, nota 2.3
- Maria, nota 6
- Carol, nota 5
- João, nota 8
- Ana, nota 8

E os predicados necessários para que ele seja capaz de responder:

- Quais são os nomes de todos os alunos?
- Quem tirou nota maior que 4?
- Quem tirou a maior nota?
- Qual foi a média de notas?
- Quem tirou nota menor que a média?
- Quem tirou nota maior que a média?

32. Considere o programa a seguir:

```
animal(cão).
animal(canário).
animal(cobra).
animal(morcego).
animal(gaivota).
voa(canário).
voa(morcego).
voa(gaivota).
dif(X,X) :- !, fail.
dif(_,_).
pássaro(X) :- animal(X), voa(X), dif(X,morcego).
```

Desenhe a árvore de busca necessária para responder a consulta:

```
?- pássaro(X).
```

Em seguida, execute o programa para ver se as respostas do sistema correspondem àquelas que você encontrou.

33. Supondo que a base de dados esteja inicialmente vazia, indique qual será o seu conteúdo após terem sido executadas as seguintes consultas:

```
?- asserta( metal(ferro) ).
?- assertz( metal(cobre) ).
?- asserta( metal(ouro) ).
?- assertz( metal(zinco) ).
?- retract( metal(X) ).
```

34. Implemente os predicados liga, desliga e lâmpada para que eles funcionem conforme indicado pelos exemplos a seguir:

```
?- liga, lâmpada(X).
X = acessa
?- desliga, lâmpada(X).
X = apagada
```

35. Crie um Sistema de Produção capaz de movimentar o cavalo de X para Y em um tabuleiro de 5x5. Ex:

```
movaCavalo(Origem, Destino, Movimentos).
?- movaCavalo(1, 2, M).
M = [1, 10, 8, 2].
```

36. Represente as seguintes informações:
    - O cachorro é um mamífero.
    - O gato é um mamífero.
    - Um mamífero é um animal.
    - O gato chama-se Netuno.

    E crie uma predicado capaz de responder se o Netuno é um animal.
37. Represente as seguintes informações:
    - Canário pode cantar.
    - Canário é amarelo.
    - Canário é um pássaro.
    - Avestruz não pode voar.
    - Avestruz é alto.
    - Avestruz é um pássaro.
    - Pássaro pode voar.
    - Pássaro tem asas.
    - Pássaro tem penas.
    - Pássaro é um animal.
    - Peixe é um animal.
    - Peixe pode nadar.
    - Animal pode respirar.
    - Animal tem pele.
    - Animal pode se mover.

    E crie os predicados necessários para responder:
    - Quais são todas as características do avestruz?
    - Quais são todas as características do canário?

      Ex:

```
todasCaracteristicas(Elem, Características).
?- todasCaracteristicas(avestruz, C).
C = [alta, asas, penas, respirar, pele, mover].
?- todasCaracteristicas(canario, C).
C = [cantar, amarelo, voar, asas, penas, respirar, pele, mover].
```