MC346 - PROJETO 4

Dado um grafo direcionado, calcular a solução do problema do caixeiro viajante, por força bruta.

O arquivo de entrada (pelo stdin) contém:

- uma lista [aresta1, aresta2, aresta3, ...] de arestas, cada aresta na forma

  ```
  [a, b, 4]
  ```

  que indica que há uma aresta do vertice de a para b e o custo desta aresta é 4. Pode ser que na lista apareça uma aresta de b para a com o mesmo ou outro custo, ou mesmo que tal aresta não exista.

- seguido do nome de um vertice, da forma

  ```
  d
  ```

  que indica que d é o vertice inicial. demais.

Escreva um programa em prolog que le do stdin a lista de arestas e o vértice inicial. O programa deve calcular qual é o caminho que sai do vértice inicial, percorre todos os vertices do grafo (seguindo as arestas) e volta para o vertice inicial que tem o custo minimo. Imprimir, usando o `print` do prolog o custo deste caminho minimo, e o caminho (na linha seguinte). O caminho deverá ser impresso na forma

```
[d, c, b, e, a, d]
```

que significa que o caminho sai de d, passa por c, b, e, a, e volta para d. Colocar um `nl` após cada `print`.

É possível que mais de um caminho seja o mais curto. Neste caso imprima apenas o caminho cuja lista é a menor na ordem padrão (`@>`) do Prolog.

É possivel que não haja nenhum caminho que sai do vertice inicial e volta para ele passando por todos os outros vertices só uma vez. Neste caso o programa deve imprimir a palavra `NADA` que é o `print` do nome/símbolo `'NADA'`.

O programa deve gerar na força bruta todas os caminhos que saem do e voltam para vertice inicial, e descobrir qual deles tem menor custo. Todos os caminhos possíveis nada mais é que todas as permutações do conjunto de vertices do grafo, removendo o vertice inicial, e depois acrescentando o vertice inicial como primeiro e último elemento dessas permutações.

Os dados de entrada estão no formato que o READ le (ou seja terminado por ponto). Assim o primeiro READ le a lista de arestas, e o segundo le o vertice

inicial. O READ coloca na saída um prompt "|:" que não é fácil remover. Assim na saída este prompt

O Prolog vai ser chamado da seguinte forma:

```
swipl -q -f prog4.pl -g main -t halt   < arqX.in
```

ou seja, seu predicado principal deve se chamar `main` e o arquivo deve se chamar `prog4.pl`.

O suzy usa o SWI Prolog 5.7.11 e nao a versao 6.X mais recente. Por exemplo, o predicado `min_member` da versao 6.X que é útil neste projeto não esta implementado na 5.X - mas não é dificil de implementar esse predicado.