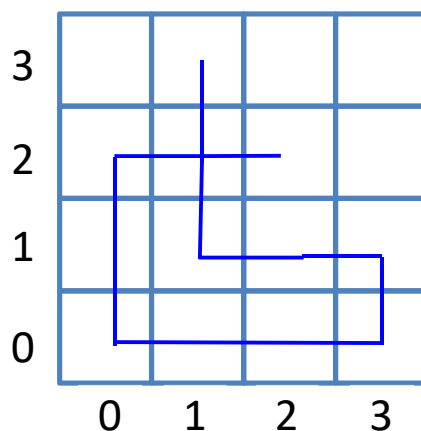


### Caracterização do problema:

Pretende-se a implementação de um programa Prolog que permita encontrar soluções num labirinto. No labirinto cada casa pode ter saídas para um máximo de quatro casas adjacentes (cima, baixo, direita, esquerda), como se mostra na figura seguinte:



As linhas a azul correspondem aos corredores do labirinto.

O labirinto da figura poderia ser representado em Prolog do seguinte modo:

base(4,4).	% dimensões x,y do labirinto
saida(1,3,b).	% da casa 1,3 pode transitar-se para a casa 1,2
saida(0,2,d).	% da casa 0,2 pode transitar-se para a casa 1,2
saida(0,2,b).	% da casa 0,2 pode transitar-se para a casa 0,1
saida(1,2,b).	% da casa 1,2 pode transitar-se para a casa 1,1
saida(1,2,d).	% da casa 1,2 pode transitar-se para a casa 2,2
saida(0,1,b).	% da casa 0,1 pode transitar-se para a casa 0,0
saida(1,1,d).	% da casa 1,1 pode transitar-se para a casa 2,1
saida(2,1,d).	% da casa 2,1 pode transitar-se para a casa 3,1
saida(3,1,b).	% da casa 3,1 pode transitar-se para a casa 3,0
saida(0,0,d).	% da casa 0,0 pode transitar-se para a casa 1,0
saida(1,0,d).	% da casa 1,0 pode transitar-se para a casa 2,0
saida(2,0,d).	% da casa 2,0 pode transitar-se para a casa 3,0

As transições são bidirecionais.

Nota: a interface gráfica que é fornecida inverte a coordenada Y.

### Parte 1 (obrigatório):

Construa no ficheiro `solução.pl` o predicado **resolver(Xp,Yp,Xc,Yc,P)** que produz, por retrocesso, em **P** as possibilidades de caminhos no labirinto entre os pontos de partida (Xp,Yp) e de chegada (Xc,Yc), aceitando-se as soluções por uma ordem qualquer.

Exemplo: ? resolver(0,2,2,1,Sol).

Sol=[e(0,2),e(0,1), e(0,0), e(1,0), e(2,0), e(3,0),e(3,1),e(2,1)]

### Parte 2 (facultativo):

Reconstrua o predicado **resolver(Xp,Yp,Xc,Yc,P)**, de forma a que a primeira solução seja a que é potencialmente mais curta.

Exemplo: ? resolver(0,2,2,1,Sol).

Sol=[e(0,2),e(1,2),e(1,1),e(2,1)]

Nota: Quem realizar a parte 2 não tem de realizar a parte 1.

### Material fornecido:

São fornecidos os seguintes elementos:

**solucao.pl** – ficheiro que deve ser usado pelos alunos para realizarem o trabalho;

**labirinto.pl** – ficheiro para testes simples;

**labirintoGrande.pl** – ficheiro para testes mais complexos;

Interface gráfica elaborada em java e Prolog.

### Prazo de entrega:

Data limite para entrega: 2020/01/06 às 23H50.