

3ª Prova de Algoritmos e Estruturas de Dados I

26/06/2015

- P: O que será corrigido?
R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, a indentação, o uso equilibrado de comentários no código e, evidentemente, a clareza. A modularidade, o correto uso de funções e procedimentos, incluindo passagem de parâmetros e o bom uso de variáveis locais e globais serão especialmente observados.
- **Introdução:** Suponha que você receba o mapa de um labirinto, no qual existem alguns caminhos que levam da entrada dele para a saída e, como em todo labirinto, alguns caminhos que são becos sem saída. O problema é não apenas encontrar um caminho que leva à saída, mas determinar o menor caminho existente.

A seguir está um trecho de código em *Pascal* que resolve este problema usando um algoritmo clássico encontrado na literatura da computação. O código apresenta algumas declarações de constantes, tipos e variáveis bem como um programa principal que será detalhado em seguida.

```
Program labirinto;  
Const
```

```
MAX = 100;  
PAREDE = -1;  
VAZIO = 0;  
LIN = 1;  
COL = 2;
```

```
Type
```

```
labirinto = array [1..MAX, 1..MAX] of integer;  
coordenada = array [1..2] of integer;  
fila = array [1..MAX] of coordenada;
```

```
Var
```

```
L: labirinto;  
tamL, entrada, saida, elem, vizinho: coordenada;  
F: fila; tamF: integer;  
distancia: integer;
```

```
Begin
```

```
iniciaFila(F, tamF);  
distancia := 1;  
lerLabirinto(L, tamL, entrada, saida);  
marcaElemLabirinto(L, saida, distancia);  
insereElemFila (F, tamF, saida);
```

```
repeat
```

```
  begin  
  retiraElemFila(F, tamF, elem);  
  distancia := valorElemLabirinto(L, elem) + 1;  
  "para cada vizinho a ser marcado de elem"
```

```
  begin
```

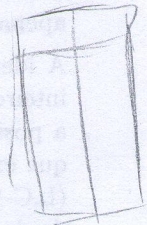
```
    marcaElemLabirinto(L, vizinho, distancia);  
    insereElemFila(F, tamF, vizinho);
```

```
  end;
```

```
until (filaVazia(F, tamF)); (tamF = 0)
```

```
imprimeMenorCaminho(L, entrada);
```

```
End.
```



l, c, el, ec, sl, sc, a: integer;