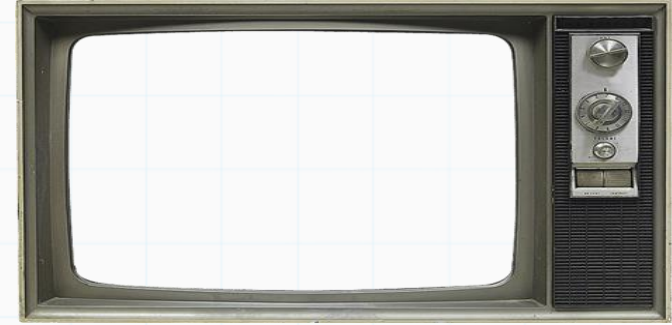


Programação Estruturada

Professor : Yuri Frota

yuri@ic.uff.br

```
int *vet;  
  
// aloca vetor de 10 inteiros  
vet = (int*) malloc( 10 * sizeof(int));  
  
float **mat;  
  
mat = (float**) malloc(l*sizeof(float*));  
  
for (int i=0; i<l; i++)  
    mat[i] = (float*) malloc(c*sizeof(float));  
  
imprime_mat(mat, l, c);
```



Ponteiros - LAB

Exemplo:

Matriz M:

```
0 4 9 3
5 0 1 400
2 1 0 8
7 5 2 0
```

k:

8

forneça itinerario:

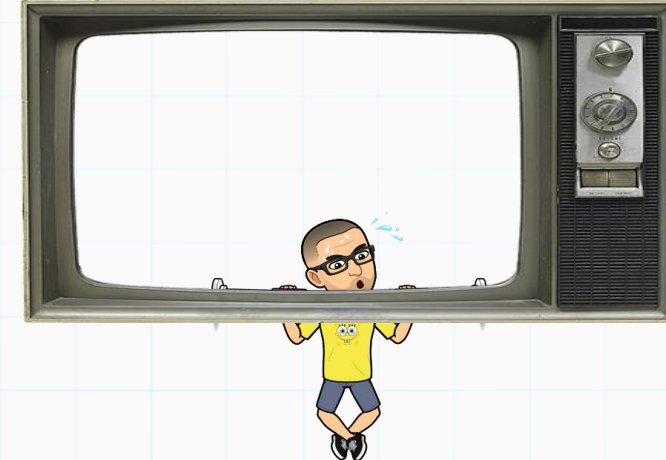
```
cidade 1:0
cidade 2:3
cidade 3:1
cidade 4:3
cidade 5:3
cidade 6:2
cidade 7:1
cidade 8:0
```

custo do itinerario = 416:

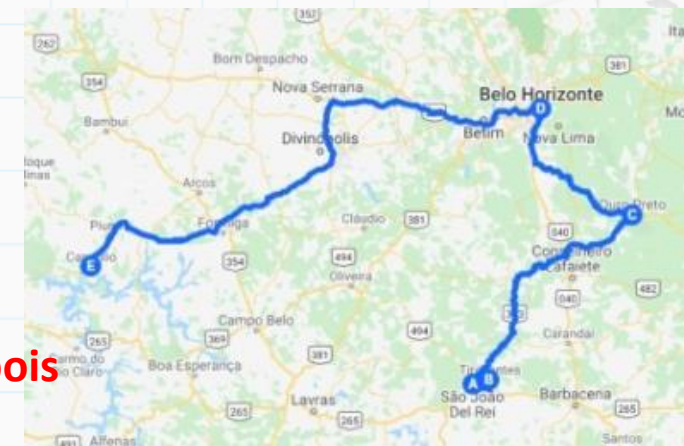
0	4	9	3
5	0	1	400
2	1	0	8
7	5	2	0

//Podemos deixar a matriz fixa para facilitar os testes

```
M[0][0] = 0; M[0][1] = 4; M[0][2] = 9; M[0][3] = 3;
M[1][0] = 5; M[1][1] = 0; M[1][2] = 1; M[1][3] = 400;
M[2][0] = 2; M[2][1] = 1; M[2][2] = 0; M[2][3] = 8;
M[3][0] = 7; M[3][1] = 5; M[3][2] = 2; M[3][3] = 0;
```



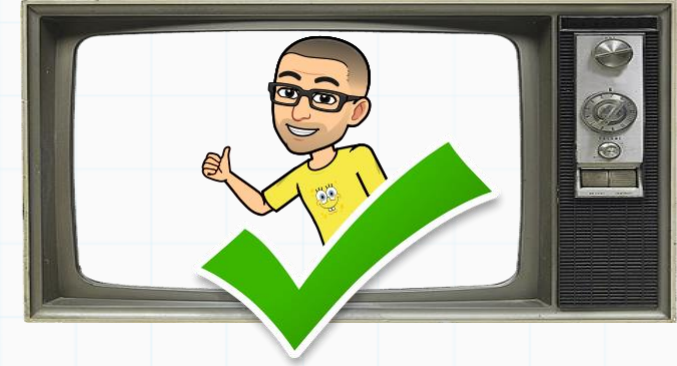
copia & cola



Use só o que aprendemos até hoje

Todos os vetores e matrizes usados no programa devem ser alocados dinamicamente e depois liberados no fim do programa.

Ponteiros - LAB



2) Palavras Cruzadas: Um jogo de palavras cruzadas pode ser representado por uma matriz $An \times m$ onde cada posição da matriz corresponde a um quadrado do jogo, sendo que 0 indica um quadrado branco e -1 indica um quadrado preto. Exemplo:

0	-1	0	-1	-1	0	-1	0
0	0	0	0	-1	0	0	0
0	0	-1	-1	0	0	-1	0
-1	0	0	0	0	-1	0	0
0	0	-1	0	0	0	-1	-1

Queremos saber quantas palavras são formadas no jogo. Para isso precisamos indicar nesta matriz as posições que são início de palavras horizontais e/ou verticais nos quadrados correspondentes, considerando que uma palavra deve ter pelo menos duas letras.

1	2	3	4		5	6
14					15	
17				18		
20						
		23			24	25
			27			

Ponteiros - LAB



2) Palavras Cruzadas: Um jogo de palavras cruzadas pode ser representado por uma matriz $An \times m$ onde cada posição da matriz corresponde a um quadrado do jogo, sendo que 0 indica um quadrado branco e -1 indica um quadrado preto. Exemplo:

0	-1	0	-1	-1	0	-1	0
0	0	0	0	-1	0	0	0
0	0	-1	-1	0	0	-1	0
-1	0	0	0	0	-1	0	0
0	0	-1	0	0	0	-1	-1

Queremos saber quantas palavras são formadas no jogo. Para isso precisamos indicar nesta matriz as posições que são início de palavras horizontais e/ou verticais nos quadrados correspondentes, considerando que uma palavra deve ter pelo menos duas letras.

0	-1	0	-1	-1	0	-1	0
0	0	0	0	-1	0	0	0
0	0	-1	-1	0	0	-1	0
-1	0	0	0	0	-1	0	0
0	0	-1	0	0	0	-1	-1

Veja que os quadrados pintados são posições de inícios de palavras horizontais (da esquerda para direita e pintados de **verde**), inícios de palavras verticais (de cima para baixo e pintados de **amarelo**), ou inícios tanto de palavras verticais quanto horizontais (pintadas de **vermelho**)

Por exemplo:

para um quadrado ser início de palavra horizontal (verde) ele precisa que:

- 1) a posição (quadrado) a esquerda não seja 0 (pois ele tem que ser a primeira letra)
- 2) e a posição (quadrado) a direita precisa ser 0 (pois a palavra tem tamanho no

mínimo 2)

Uma intuição semelhante pode ser feita com as palavras verticais.

1	2	3	4	-1	5	6	
14				-1	15		
17				18			
20					-1	-1	
			23			24	25
				27			

Ponteiros - LAB



2) Palavras Cruzadas: Um jogo de palavras cruzadas pode ser representado por uma matriz $An \times m$ onde cada posição da matriz corresponde a um quadrado do jogo, sendo que 0 indica um quadrado branco e -1 indica um quadrado preto. Exemplo:

0	-1	0	-1	-1	0	-1	0
0	0	0	0	-1	0	0	0
0	0	-1	-1	0	0	-1	0
-1	0	0	0	0	-1	0	0
0	0	-1	0	0	0	-1	-1

Queremos saber quantas palavras são formadas no jogo. Para isso precisamos indicar nesta matriz as posições que são início de palavras horizontais e/ou verticais nos quadrados correspondentes, considerando que uma palavra deve ter pelo menos duas letras.

0	-1	0	-1	-1	0	-1	0
0	0	0	0	-1	0	0	0
0	0	-1	-1	0	0	-1	0
-1	0	0	0	0	-1	0	0
0	0	-1	0	0	0	-1	-1

Veja que os quadrados pintados são posições de inícios de palavras horizontais (da esquerda para direita e pintados de **verde**), inícios de palavras verticais (de cima para baixo e pintados de **amarelo**), ou inícios tanto de palavras verticais quanto horizontais (pintadas de **vermelho**)

1	-1	2	-1	-1	3	-1	4
5	6	0	0	-1	7	0	0
8	0	-1	-1	9	0	-1	0
-1	10	0	11	0	-1	12	0
13	0	-1	14	0	0	-1	-1

Queremos fazer um programa que identifique essas casas e as numere sequencialmente da esquerda para direita e de cima para baixo (substituindo os '0's) e transformando a matriz.

Veja um exemplo de execução a seguir:

1	2	3	4	-1	5	6	
14				-1	15		
17				18			
20						-1	
			23			24	25
				27			

Ponteiros - LAB



Exemplo:

Matriz =

```
0 -1 0 -1 -1 0 -1 0
0 0 0 0 -1 0 0 0
0 0 -1 -1 0 0 -1 0
-1 0 0 0 0 -1 0 0
0 0 -1 0 0 0 -1 -1
```

Matriz processada

```
1 -1 2 -1 -1 3 -1 4
5 6 0 0 -1 7 0 0
8 0 -1 -1 9 0 -1 0
-1 10 0 11 0 -1 12 0
13 0 -1 14 0 0 -1 -1
```

Use só o que aprendemos até hoje

Todos os vetores e matrizes usados no programa devem ser alocados dinamicamente e depois liberados no fim do programa.

0	-1	0	-1	-1	0	-1	0
0	0	0	0	-1	0	0	0
0	0	-1	-1	0	0	-1	0
-1	0	0	0	0	-1	0	0
0	0	-1	0	0	0	-1	-1

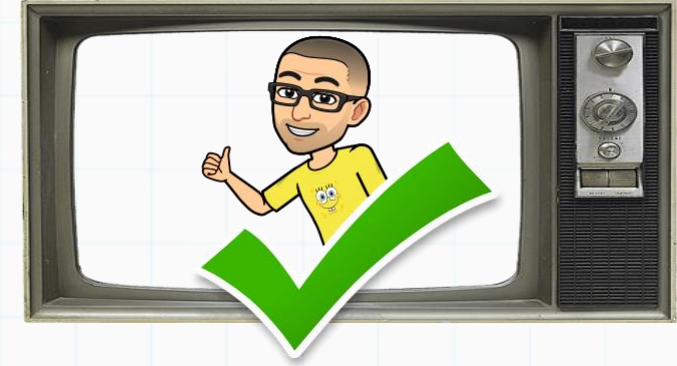
//Podemos deixar a matriz fixa para facilitar os testes

```
M[0][0] = 0; M[0][1] = -1; M[0][2] = 0; M[0][3] = -1; M[0][4] = -1; M[0][5] = 0; M[0][6] = -1; M[0][7] = 0;
M[1][0] = 0; M[1][1] = 0; M[1][2] = 0; M[1][3] = 0; M[1][4] = -1; M[1][5] = 0; M[1][6] = 0; M[1][7] = 0;
M[2][0] = 0; M[2][1] = 0; M[2][2] = -1; M[2][3] = -1; M[2][4] = 0; M[2][5] = 0; M[2][6] = -1; M[2][7] = 0;
M[3][0] = -1; M[3][1] = 0; M[3][2] = 0; M[3][3] = 0; M[3][4] = 0; M[3][5] = -1; M[3][6] = 0; M[3][7] = 0;
M[4][0] = 0; M[4][1] = 0; M[4][2] = -1; M[4][3] = 0; M[4][4] = 0; M[4][5] = 0; M[4][6] = -1; M[4][7] = -1;
```

1	2	3	4	5	6
14				15	
17			18		
20					
		23		24	25
			27		

copia & cola

Ponteiros - LAB



Exemplo 2:

Matriz =

```
-1 -1 0 -1 -1 0 -1 -1
0 0 -1 0 -1 0 0 0
0 0 0 -1 0 0 -1 0
-1 -1 0 0 0 -1 0 0
0 0 -1 0 -1 0 -1 0
0 0 -1 0 -1 0 0 0
0 0 -1 0 -1 0 -1 0
```

Matriz processada

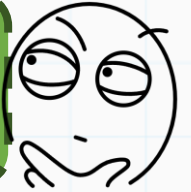
?

copia & cola

//Podemos deixar a matriz fixa para facilitar os testes

```
M[0][0] = -1; M[0][1] = -1; M[0][2] = 0; M[0][3] = -1; M[0][4] = -1; M[0][5] = 0; M[0][6] = -1; M[0][7] = -1;
M[1][0] = 0; M[1][1] = 0; M[1][2] = -1; M[1][3] = 0; M[1][4] = -1; M[1][5] = 0; M[1][6] = 0; M[1][7] = 0;
M[2][0] = 0; M[2][1] = 0; M[2][2] = 0; M[2][3] = -1; M[2][4] = 0; M[2][5] = 0; M[2][6] = -1; M[2][7] = 0;
M[3][0] = -1; M[3][1] = -1; M[3][2] = 0; M[3][3] = 0; M[3][4] = 0; M[3][5] = -1; M[3][6] = 0; M[3][7] = 0;
M[4][0] = 0; M[4][1] = 0; M[4][2] = -1; M[4][3] = 0; M[4][4] = -1; M[4][5] = 0; M[4][6] = -1; M[4][7] = 0;
M[5][0] = 0; M[5][1] = 0; M[5][2] = -1; M[5][3] = 0; M[5][4] = -1; M[5][5] = 0; M[5][6] = 0; M[5][7] = 0;
M[6][0] = 0; M[6][1] = 0; M[6][2] = -1; M[6][3] = 0; M[6][4] = -1; M[6][5] = 0; M[6][6] = -1; M[6][7] = 0;
```

Quanto é a soma da primeira coluna da matriz processada nesse segundo exemplo ?



Ponteiros - LAB

Use só o que aprendemos até hoje

3) A nave espacial **USS Enterprise** é movida por um reator formado por uma grade de cristais energéticos. Esse reator é representado por uma matriz n x m, em que cada posição contém um número inteiro entre **0** e **10**, indicando o nível de energia de um cristal. A energia dos cristais evolui em rodadas, de acordo com as regras abaixo:

Regra 1 – Cristais superaquecidos:

Se o valor de um cristal for **maior ou igual a 8**, ele perde **2** unidades de energia por superaquecimento.

2	9	3
5	8	1
4	2	7

 →

2	9	3
5	6	1
4	2	7

Regra 2 – Cristais fracos:

Se o valor de um cristal for **menor ou igual a 3**, ele absorve energia dos vizinhos fortes. Conte quantos vizinhos ortogonais (cima, baixo, esquerdo, direito) possuem valor **maior ou igual a 7** e some +1 por vizinho forte.

2	9	3
5	2	1
4	7	7

 →

2	9	3
5	4	1
4	7	7

Regra 3 – Cristais intermediários

Se o valor do cristal estiver entre **4 e 7**, inclusive, faça o seguinte:
se ele tiver **2 ou mais vizinhos fortes** (valor maior ou igual a 7), então soma +1 de energia para ele
se ele tiver **0 vizinhos fortes**, então diminua 1 de energia para ele
caso contrário (**caso tenha apenas 1 vizinho forte**), ele permanece inalterado.

2	9	3
8	2	1
4	2	7

 →

3	9	3
8	4	1
4	2	6

Regra 4 – Limites de energia

Após a atualização:

se o novo valor ficar menor que 0, ele deve passar a valer 0
se o novo valor ficar maior que 10, ele deve passar a valer 10

2	2	3
1	0	3
4	0	7

 →

2	2	3
1	-1	3
4	0	7

 →

2	2	3
1	0	3
4	0	7

continua



Ponteiros - LAB



Regra 5 -
Todas as posições da matriz devem ser atualizadas **simultaneamente** em cada rodada.
Portanto, você **não deve alterar diretamente a matriz original durante o cálculo da rodada.**
Use uma **matriz auxiliar** para armazenar os novos valores (depois copie a matriz alterada de volta para a matriz original).

Como fica a matriz de cristais energéticos depois de 5 rodadas passadas ?

Exemplo
n=3 e
m=3:

2	9	3
5	8	1
4	2	7

copia & cola

//Podemos deixar a matriz fixa para facilitar os testes

```
A[0][0] = 2; A[0][1]= 9; A[0][2]=3;  
A[1][0] = 5; A[1][1]= 8; A[1][2]=1;  
A[2][0] = 4; A[2][1]= 2; A[2][2]=7;
```

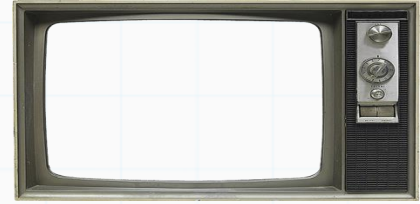
```
MATRIZ  
2 9 3  
5 8 1  
4 2 7  
  
rodada 1:  
3 7 4  
5 6 3  
3 4 6  
  
rodada 2:  
4 6 4  
4 6 3  
3 3 5  
  
rodada 3:  
3 5 3  
3 5 3  
3 3 4  
  
rodada 4:  
3 4 3  
3 4 3  
3 3 3  
  
rodada 5:  
3 3 3  
3 3 3  
3 3 3
```

continua

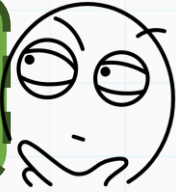
Todos os vetores e matrizes usados no programa devem ser alocados dinamicamente e depois liberados no fim do programa.



Ponteiros - LAB



Para a seguinte matriz energética, qual é a soma da primeira linha da matriz na quinta (5) rodada ?



copia & cola

//Podemos deixar a matriz fixa para facilitar os testes

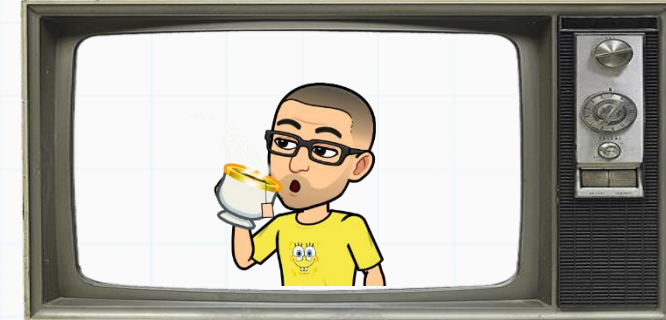
MATRIZ

```
0  9  1  8  2 10
8  2  9  1  8  0
1 10  3  9  2  8
9  0  8  2 10  1
2  8  1  9  3 10
10 1  9  0  8  2
```

```
A[0][0] = 0; A[0][1] = 9; A[0][2] = 1; A[0][3] = 8; A[0][4] = 2; A[0][5] = 10;
A[1][0] = 8; A[1][1] = 2; A[1][2] = 9; A[1][3] = 1; A[1][4] = 8; A[1][5] = 0;
A[2][0] = 1; A[2][1] = 10; A[2][2] = 3; A[2][3] = 9; A[2][4] = 2; A[2][5] = 8;
A[3][0] = 9; A[3][1] = 0; A[3][2] = 8; A[3][3] = 2; A[3][4] = 10; A[3][5] = 1;
A[4][0] = 2; A[4][1] = 8; A[4][2] = 1; A[4][3] = 9; A[4][4] = 3; A[4][5] = 10;
A[5][0] = 10; A[5][1] = 1; A[5][2] = 9; A[5][3] = 0; A[5][4] = 8; A[5][5] = 2;
```



Ponteiros - LAB

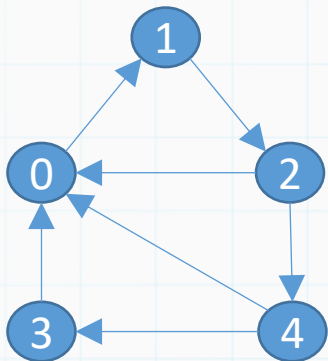


4a) Dado uma cidade c , determinar quantas estradas saem e quantas chegam à cidade c .

Veja exemplo:

Matriz M:

```
1 1 0 0 0
0 1 1 0 0
1 0 1 0 1
1 0 0 1 0
1 0 0 1 1
```



1	1	0	0	0
0	1	1	0	0
1	0	1	0	1
1	0	0	1	0
1	0	0	1	1

copia & cola

--- Menu ---

1) grau de conexao

10) Sair

opcao:1

qual cidade:0

Sai 1 e entram 3 estradas na cidade 0

--- Menu ---

1) grau de conexao

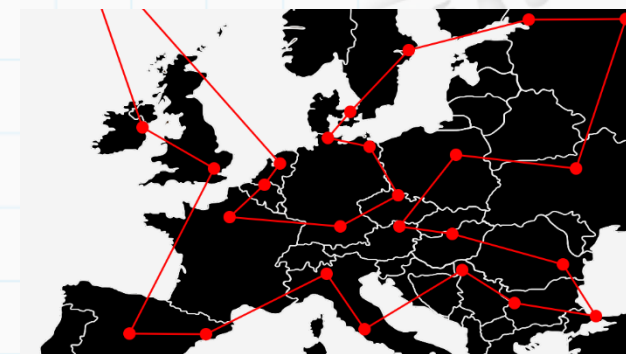
10) Sair

opcao:10

fim

//Podemos deixar a matriz fixa para facilitar os testes

```
M[0][0] = 1; M[0][1] = 1; M[0][2] = 0; M[0][3] = 0; M[0][4] = 0;
M[1][0] = 0; M[1][1] = 1; M[1][2] = 1; M[1][3] = 0; M[1][4] = 0;
M[2][0] = 1; M[2][1] = 0; M[2][2] = 1; M[2][3] = 0; M[2][4] = 1;
M[3][0] = 1; M[3][1] = 0; M[3][2] = 0; M[3][3] = 1; M[3][4] = 0;
M[4][0] = 1; M[4][1] = 0; M[4][2] = 0; M[4][3] = 1; M[4][4] = 1;
```



Ponteiros - LAB

4b) Queremos saber a qual das cidades chega o maior número de estradas? Em caso de empate, pode ser qualquer uma das de maior número de entradas.

Veja exemplo:

Matriz M:

```
1 1 0 0 0
0 1 1 0 0
1 0 1 0 1
1 0 0 1 0
1 0 0 1 1
```

--- Menu ---

```
1) grau de conexao
2) cidade facil de chegar
10) Sair
```

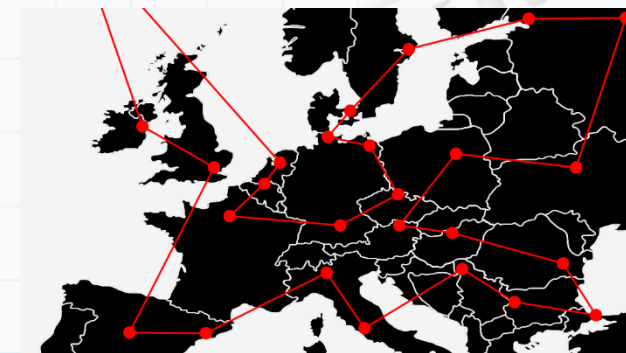
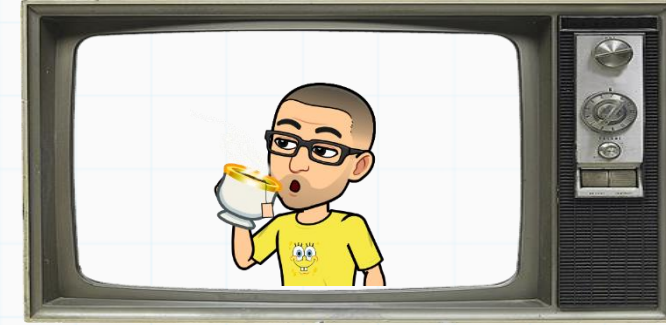
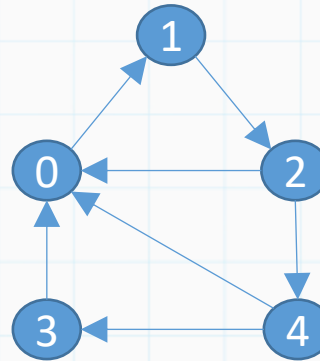
opcao:2

A cidade 0 chegam o maior numero de estradas = 3

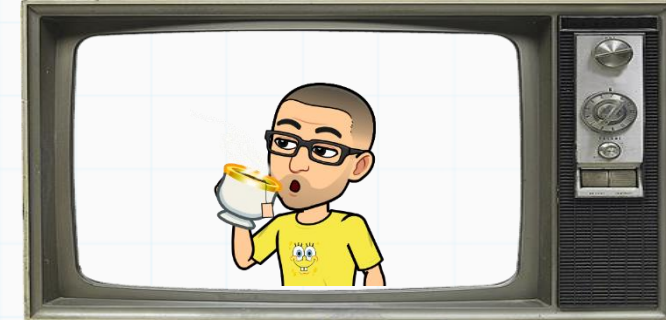
--- Menu ---

```
1) grau de conexao
2) cidade facil de chegar
10) Sair
```

opcao:10
fim



Ponteiros - LAB



4c) Dada uma sequência de k cidades, verificar se é possível realizar o roteiro correspondente

Veja exemplo:

Matriz M:

```
1 1 0 0 0
0 1 1 0 0
1 0 1 0 1
1 0 0 1 0
1 0 0 1 1
```

--- Menu ---

- 1) grau de conexao
- 2) cidade facil de chegar
- 3) roteiro possivel
- 10) Sair

opcao:3

quantas cidades no roteiro:4

```
cidade 1:1
cidade 2:2
cidade 3:4
cidade 4:0
```

-- viagem possivel :) --

--- Menu ---

- 1) grau de conexao
- 2) cidade facil de chegar
- 3) roteiro possivel
- 10) Sair

opcao:3

quantas cidades no roteiro:4

```
cidade 1:4
cidade 2:3
cidade 3:3
cidade 4:0
```

-- viagem possivel :) --

--- Menu ---

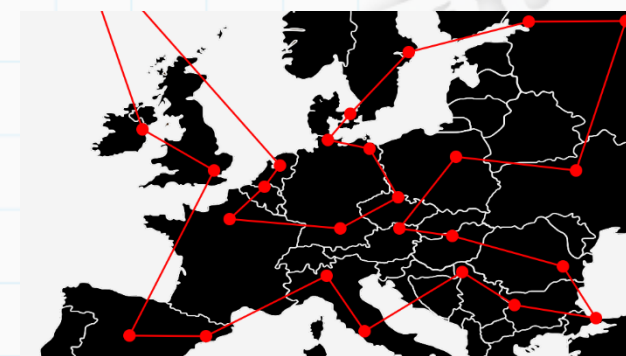
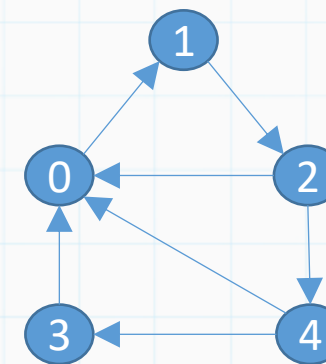
- 1) grau de conexao
- 2) cidade facil de chegar
- 3) roteiro possivel
- 10) Sair

opcao:3

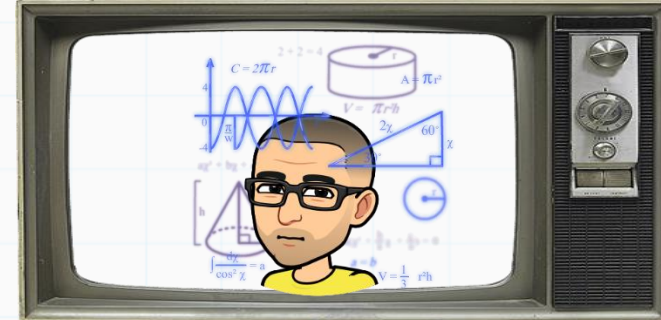
quantas cidades no roteiro:4

```
cidade 1:3
cidade 2:0
cidade 3:2
cidade 4:4
```

-- viagem impossivel :(--
nao existe estrada de 0 para 2



Ponteiros - LAB



4d) **DESAFIO**: Dadas cidades c1 e c2, determinar se é possível ir da cidade c1 para a cidade c2 e qual seria esse caminho. **DICA: Use recursão.**

Veja exemplo:

Matriz M:

```
1 1 0 0 0
0 1 1 0 0
1 0 1 0 1
1 0 0 1 0
1 0 0 1 1
```

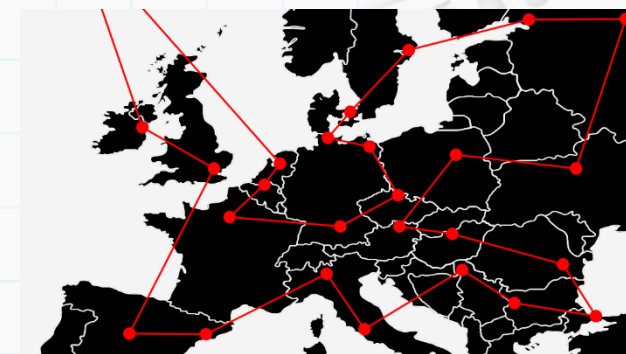
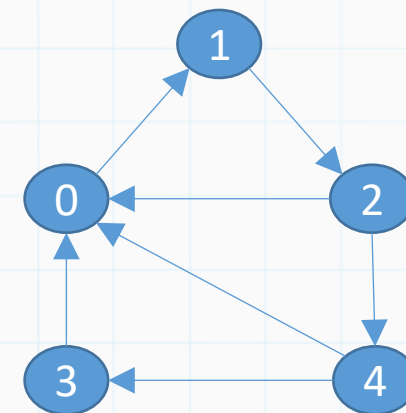
--- Menu ---

- 1) grau de conexao
- 2) cidade facil de chegar
- 3) roteiro possivel
- 4) caminho
- 10) Sair

opcao:4

origem e destino: 0 3

-- encontrou caminho = 0 -> 1 -> 2 -> 4 -> 3



Ponteiros - LAB



4d) **DESAFIO**: Dadas cidades c1 e c2, determinar se é possível ir da cidade c1 para a cidade c2 e qual seria esse caminho. **DICA**: Use recursão.

Veja exemplo:

copia & cola

Matriz M:

1	1	0	1	0	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	0	1	1	0	1	0	0
0	0	1	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	1	0	0	0	1

//Podemos deixar a matriz fixa para facilitar os testes

```
M[0][0] = 1; M[0][1] = 1; M[0][2] = 0; M[0][3] = 1; M[0][4] = 0;
M[0][5] = 0; M[0][6] = 0; M[0][7] = 0; M[0][8] = 0; M[0][9] = 0;
M[1][0] = 0; M[1][1] = 1; M[1][2] = 1; M[1][3] = 0; M[1][4] = 0;
M[1][5] = 0; M[1][6] = 1; M[1][7] = 0; M[1][8] = 0; M[1][9] = 0;
M[2][0] = 0; M[2][1] = 0; M[2][2] = 1; M[2][3] = 0; M[2][4] = 1;
M[2][5] = 0; M[2][6] = 0; M[2][7] = 0; M[2][8] = 0; M[2][9] = 0;
M[3][0] = 0; M[3][1] = 0; M[3][2] = 0; M[3][3] = 1; M[3][4] = 1;
M[3][5] = 0; M[3][6] = 0; M[3][7] = 0; M[3][8] = 0; M[3][9] = 0;
M[4][0] = 0; M[4][1] = 0; M[4][2] = 0; M[4][3] = 0; M[4][4] = 1;
M[4][5] = 1; M[4][6] = 0; M[4][7] = 1; M[4][8] = 0; M[4][9] = 0;
M[5][0] = 0; M[5][1] = 0; M[5][2] = 1; M[5][3] = 0; M[5][4] = 0;
M[5][5] = 1; M[5][6] = 1; M[5][7] = 0; M[5][8] = 0; M[5][9] = 0;
M[6][0] = 0; M[6][1] = 0; M[6][2] = 0; M[6][3] = 0; M[6][4] = 0;
M[6][5] = 0; M[6][6] = 1; M[6][7] = 1; M[6][8] = 0; M[6][9] = 0;
M[7][0] = 0; M[7][1] = 0; M[7][2] = 0; M[7][3] = 0; M[7][4] = 0;
M[7][5] = 0; M[7][6] = 0; M[7][7] = 1; M[7][8] = 1; M[7][9] = 0;
M[8][0] = 0; M[8][1] = 0; M[8][2] = 0; M[8][3] = 0; M[8][4] = 0;
M[8][5] = 0; M[8][6] = 0; M[8][7] = 0; M[8][8] = 1; M[8][9] = 1;
M[9][0] = 1; M[9][1] = 0; M[9][2] = 0; M[9][3] = 0; M[9][4] = 0;
M[9][5] = 1; M[9][6] = 0; M[9][7] = 0; M[9][8] = 0; M[9][9] = 1;
```

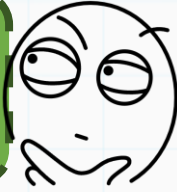
Existe caminho de 0 a 9 ? Qual ?



Recursão - LAB



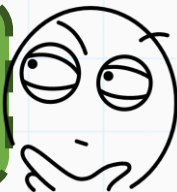
Quanto é a soma da primeira coluna da matriz processada nesse segundo exemplo ?



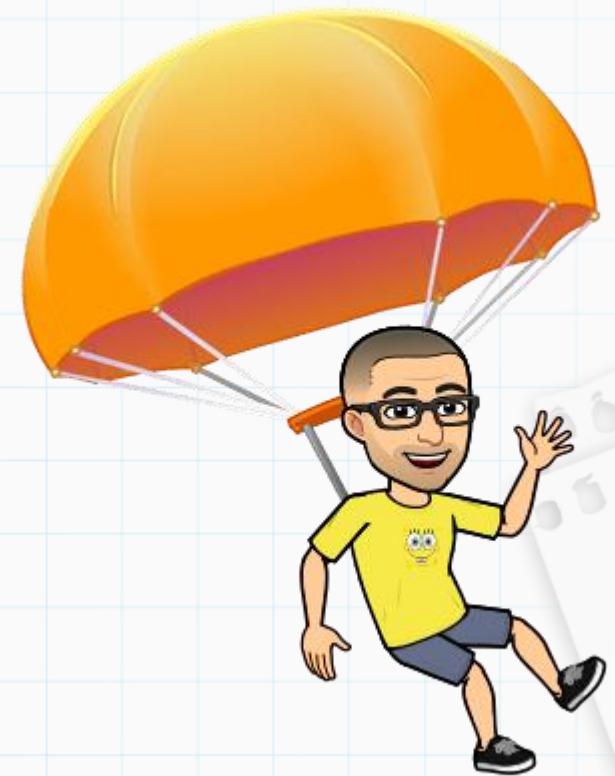
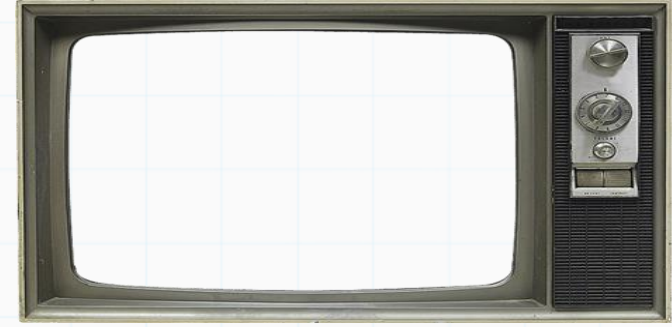
Para a seguinte matriz energética, qual é a soma da primeira linha da matriz na quinta (5) rodada ?



Existe caminho de 0 a 9 ? Qual ?



Até a próxima



Slides baseados no curso de Aline Nascimento