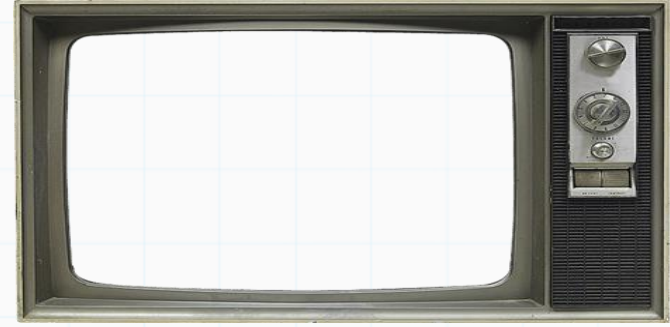


Programação Estruturada

Professor : Yuri Frota

yuri@ic.uff.br



```
int *vet;
```

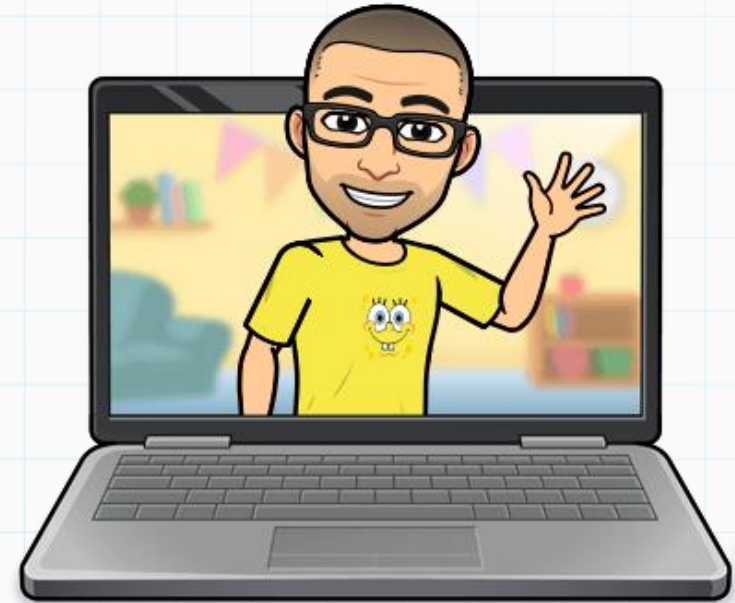
```
// aloca vetor de 10 inteiros  
vet = (int*) malloc( 10 * sizeof(int));
```

```
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

1) Itinerários: Os elementos a_{ij} de uma matriz inteira A $n \times n$ representam os custos de transporte da cidade i para a cidade j . Dados $k1$ itinerários, cada um com $k2$ cidades, calcular o custo total para cada itinerário.

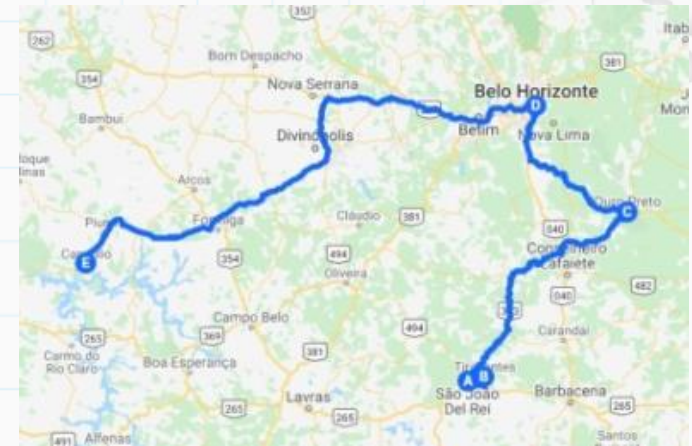
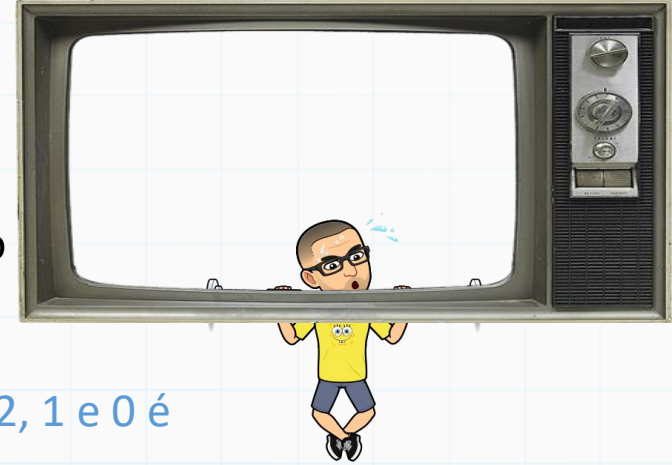
Exemplo: $n=4$ e $A=$

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

O custo do itinerário das cidades 0, 3, 1, 3, 3, 2, 1 e 0 é
 $a_{03} + a_{31} + a_{13} + a_{33} + a_{32} + a_{21} + a_{10} = 416$

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

Veja um exemplo de execução a seguir:



Ponteiros - LAB

Exemplo:

Matriz M:

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

forneça k1 e k2:

```
1
8
```

forneça itinerario 1:

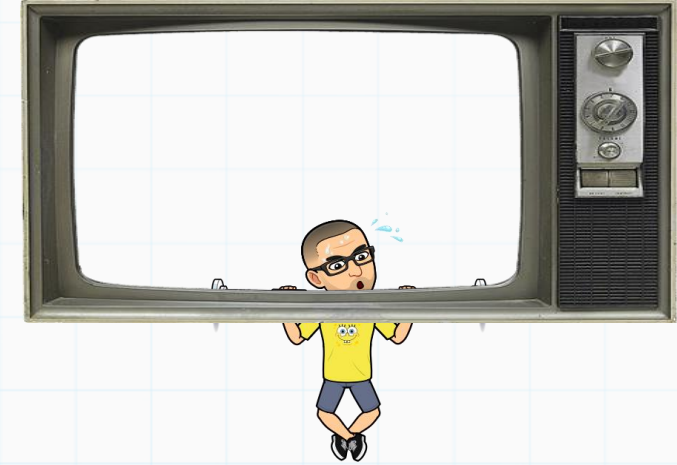
```
cidade 1:0
cidade 2:3
cidade 3:1
cidade 4:3
cidade 5:3
cidade 6:2
cidade 7:1
cidade 8:0
```

custo do itinerario = 416:

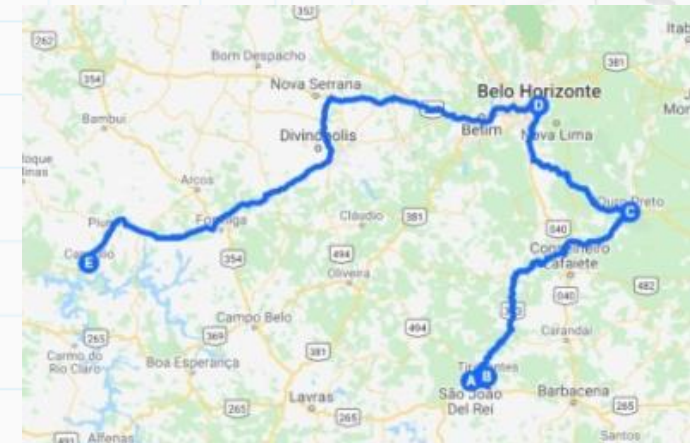
0	4	9	3
5	0	1	400
2	1	0	8
7	5	2	0

//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



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 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.

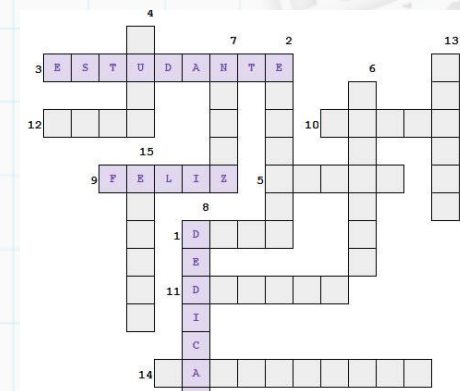
0	-1	0	-1	-1	0	-1	0
0	0	0	0	-1	0	0	0
0	0	-1	-1	0	0	-1	0
-1	0	0	0	0	-1	0	0
0	0	-1	0	0	0	-1	-1

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 (substituindo os '0's) e transformando a matriz.

Veja um exemplo de execução a seguir:



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
```

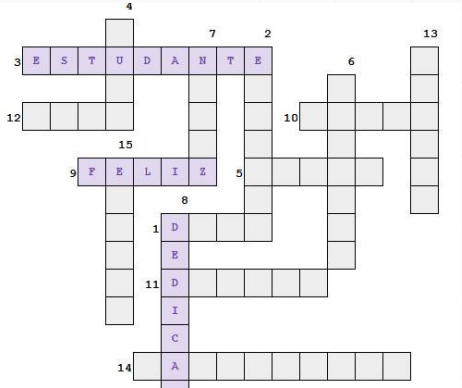
Todos os vetores e matrizes usados no programa devem ser alocados dinamicamente e depois liberados no fim do programa.

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

copia & cola

//Podemos deixar a matriz fixa para facilitar os testes

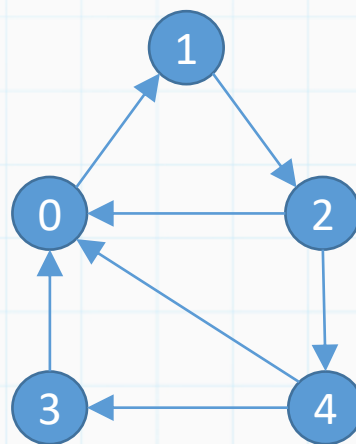
```
M[0][0] = 0; M[0][1] = -1; M[0][2] = 0; M[0][3] = -1; M[0][4] = -1; M[0][5] = 0; M[0][6] = -1; M[0][7] = 0;
M[1][0] = 0; M[1][1] = 0; M[1][2] = 0; M[1][3] = 0; M[1][4] = -1; M[1][5] = 0; M[1][6] = 0; M[1][7] = 0;
M[2][0] = 0; M[2][1] = 0; M[2][2] = -1; M[2][3] = -1; M[2][4] = 0; M[2][5] = 0; M[2][6]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



A cartoon illustration of a man with glasses and a yellow shirt, drinking from a white cup. The character is displayed on a vintage television set, which has a dark frame and a control panel on the right side with various knobs and buttons. The background within the TV screen is a light blue grid.

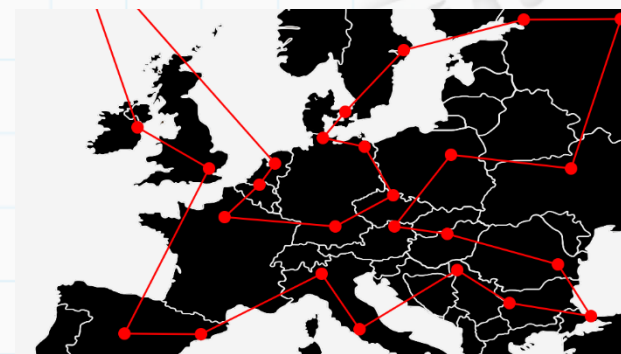
Exemplo: $n=5$ $M =$

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



Queremos fazer um programa que irá interagir com o usuário através de um menu que pode responder várias perguntas de conexidades em relação as cidades.

- Cada item dessa questão será um item do menu que deve ser implementado em uma função diferente.
- Todos os vetores e matrizes usados no programa devem ser alocados dinamicamente e depois liberados no fim do programa. A alocação e a liberação devem ser feitas por funções também.
- Deve-se usar uma estrutura “switch” para escolher que função será realizada, de acordo com a escolha do usuário



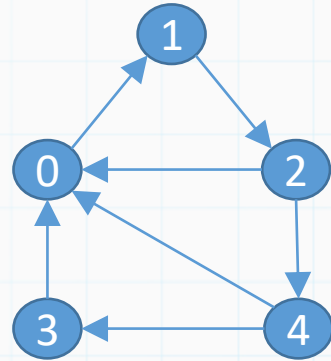
Ponteiros - LAB

3a) 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

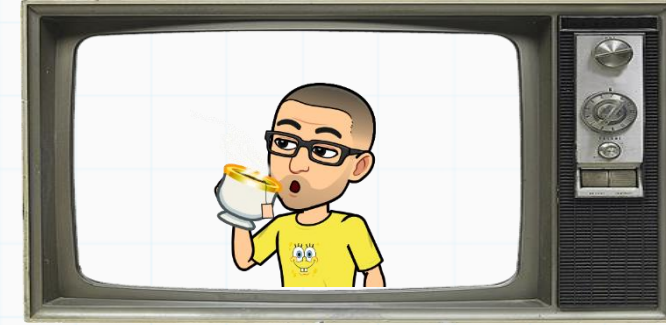
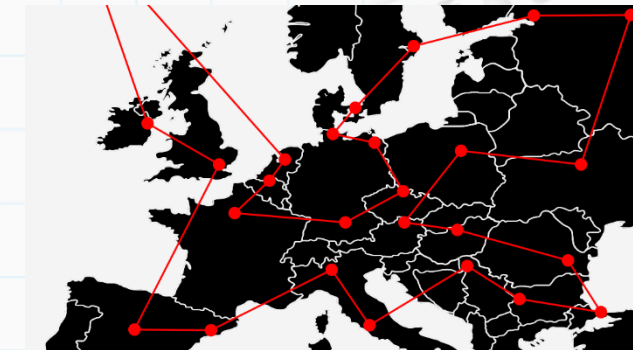
10) Sair

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

3b) 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

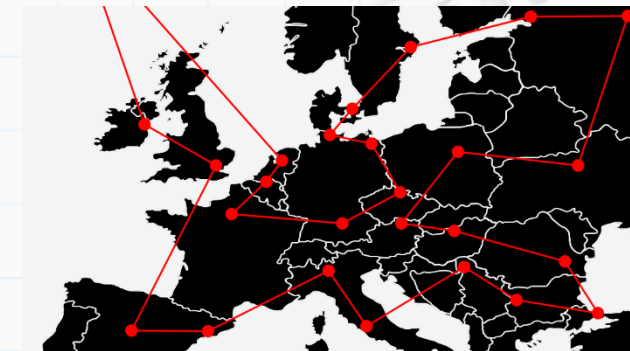
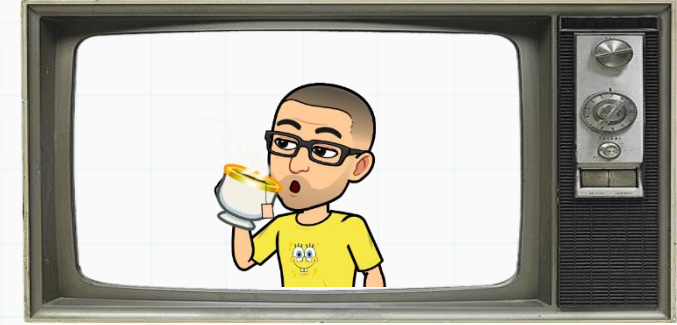
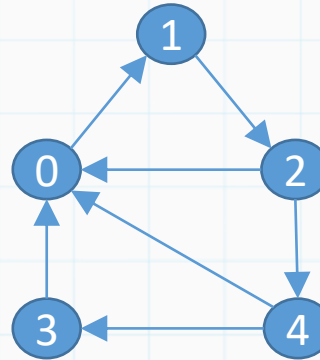
A cidade 0 chegam o maior numero de
estradas = 3

--- Menu ---

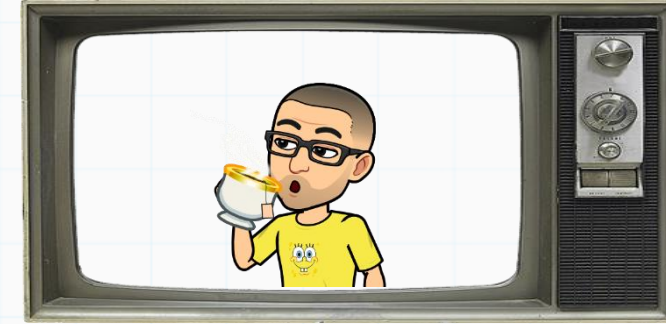
1) grau de conexao
2) cidade facil de chegar
10) Sair

opcao:10

fim



Ponteiros - LAB



3c) 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

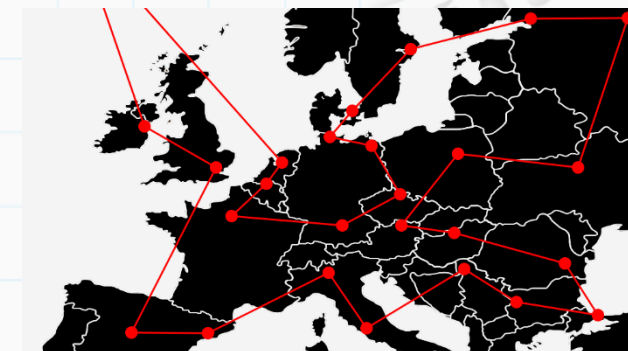
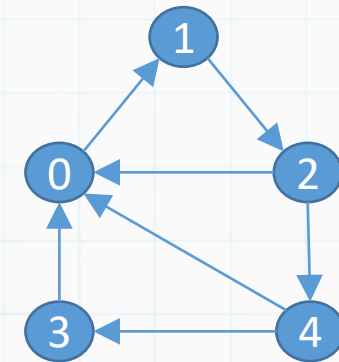
opcao:3

quantas cidades no roteiro:4

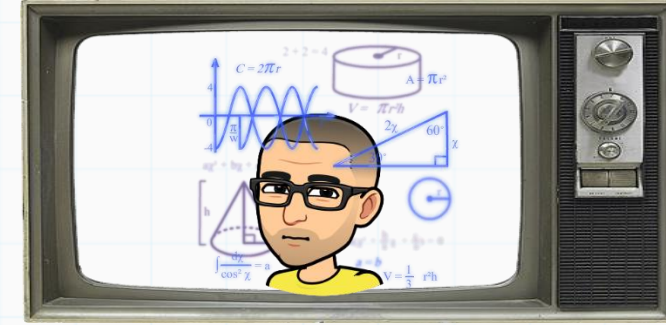
cidade 1:3
cidade 2:0
cidade 3:2
cidade 4:4

-- viagem impossivel :(--

nao existe estrada de 0 para 2



Ponteiros - LAB



3d) **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

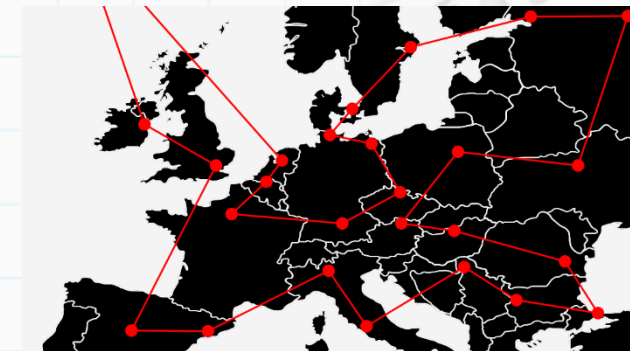
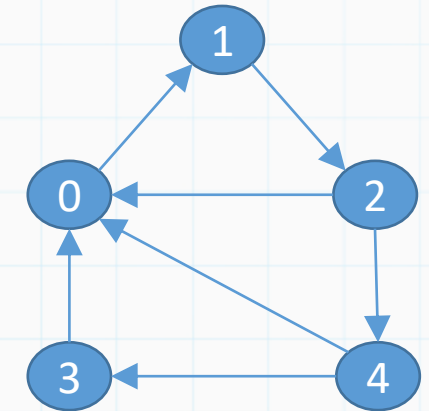
--- Menu ---

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

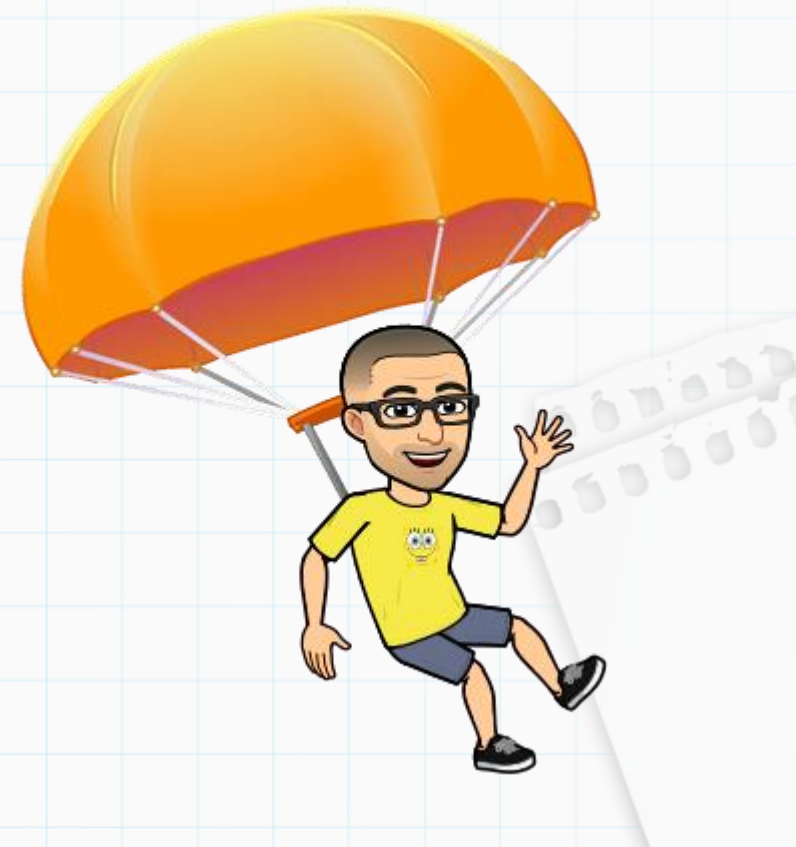
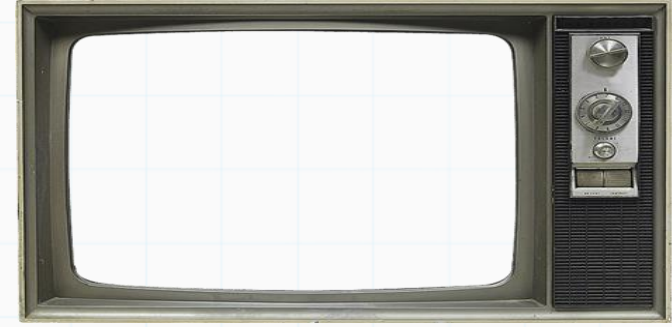
opcao:4

origem e destino: 0 3

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



Até a próxima



Slides baseados no curso de Aline Nascimento