Algoritmia e Programação

Trabalho Prático (2023-2024)

Enunciado

distâncias entre cidades

código da cidade		Porto	Aveiro	Braga	Coimbra	Lisboa	Fátima
(0)	Porto	0	50	60	130	300	200
(1)	Aveiro	50	0	130	70	250	140
(2)	Braga	60	130	0	180	370	250
(3)	Coimbra	130	70	180	0	200	90
(4)	Lisboa	300	250	370	200	0	130
(5)	Fátima	200	140	250	90	130	0

A empresa TravelDEI organiza viagens de autocarro com rotas que percorrem diversas cidades portuguesas, permanecendo pelo menos 1 dia em cada cidade, para os turistas terem a oportunidade de conhecer e desfrutar dessas cidades.
Algumas características da empresa são:

- Para já, possui um alvará que lhe permite viajar entre as seguintes cidades: Porto, Aveiro, Braga, Coimbra, Fátima e Lisboa;
- Possui uma frota variável de autocarros (Bus0, Bus1, …) de acordo com as necessidades. A numeração dos autocarros é sequencial (0,1,2,…,n) após a palavra "Bus";
- Cada autocarro possui uma rota própria, podendo ser diferente da dos outros autocarros.

A empresa efetua um planeamento com as rotas previstas para os vários autocarros. O planeamento é representado através de uma matriz, com dimensões LxC, em que L e C representam a quantidade de linhas e colunas da matriz, respetivamente. Cada linha da matriz representa um autocarro e as colunas representam dias de viagem do planeamento, contendo um número inteiro com o código da cidade onde o respetivo autocarro estará nesse dia.

Por exemplo, a seguinte matriz representa o planeamento para os autocarros Bus0 e Bus1 para 3 dias:

0	4	0
2	0	1

O autocarro "Bus0" estará no Porto (0), Lisboa (4) e Porto (0), nos dias 0,1 e 2, respetivamente.
O autocarro "Bus1" estará em Braga (2), Porto (0) e Aveiro (1), nos dias 0,1 e 2, respetivamente.

Considere-se que as distâncias entre cidades são as representadas na tabela acima.

Pretende-se efetuar algumas operações de análise e manipulação da informação da matriz para obter algumas estatísticas e facilitar uma adequada tomada de decisões no planeamento de futuras viagens.

Um planeamento é definido pela seguinte estrutura:
- 1ª linha – texto descritivo do planeamento e data inicio ;
- 2ª linha - dois inteiros (L e C), separados por um espaço, indicando a quantidade de autocarros (L) e a quantidade de dias do planeamento (C);
- L linhas, cada uma contendo C números inteiros representativos do código das cidades onde estará esse autocarro em cada dia, separados por um espaço.

Exemplo de um planeamento:

```
Passagem de ano;2023/12/28
3 6
0 1 3 5 4 0
0 2 3 4 4 0
4 5 3 0 4 0
```

Com o objetivo de responder aos requisitos deste trabalho, recorra às boas práticas lecionadas e implemente um programa em Java (sem interação com o utilizador) com as seguintes funcionalidades:

Algoritmia e Programação

Trabalho Prático (2023-2024)

a) Ler a informação de um planeamento e armazená-la em memória numa matriz (evitar variáveis globais);
b) Visualizar a matriz de planeamento no ecrã;
c) Obter uma matriz com os km a percorrer, em cada dia, para cada autocarro, e visualizar essa matriz com os valores das colunas alinhados à direita;
d) Obter um array com o total de km a percorrer por cada autocarro e visualizar esse array;
e) Visualizar a totalidade de kms a percorrer pela frota de autocarros;
f) Visualizar o máximo de km a percorrer num dia e em que dias isso acontece;
g) Visualizar os autocarros que estarão na mesma cidade mais de 1 dia consecutivo;
h) Visualizar o dia mais tardio e a cidade, em que todos os autocarros estarão na mesma cidade;
i) Visualizar um histograma para representar os kms a percorrer pelos autocarros (em %). Cada linha do histograma deverá ter no máximo 10 asteriscos (*).
j) Visualizar qual é o autocarro mais próximo de outro autocarro num determinado dia. O autocarro e o dia são especificados na entrada de dados, a seguir ao planeamento (ex: 0 3 – Bus0 no dia 3).

OBS: O programa deve executar de forma sequencial todas as alíneas e mostrar no ecrã o respetivo resultado, de forma idêntica ao exemplo seguinte:

Exemplo:

Input	output
Passagem de ano;2023/12/28 3 6 0 1 3 5 4 0 0 2 3 4 4 0 4 5 3 0 4 0 0 3	**b)** **Bus0 : 0 1 3 5 4 0** **Bus1 : 0 2 3 4 4 0** **Bus2 : 4 5 3 0 4 0** **c)** **Bus0 : 0 50 70 90 130 300** **Bus1 : 0 60 180 200 0 300** **Bus2 : 0 130 90 130 300 300** **d)** **Bus0 : 640 km** **Bus1 : 740 km** **Bus2 : 950 km** **e)** **Total de km a percorrer pela frota = 2330 km** **f)** **máximo de kms num dia: (900 km), dias: [5]** **g)** **Autocarros que permanecem mais de 1 dia consecutivo na mesma cidade: Bus1** **h)** **No dia <5>, todos os autocarros estarão em <Porto>** **i)** **Bus0 (27.47%) :**** **Bus1 (31.76%) :***** **Bus2 (40.77%) :****** **j)** **No dia <3>, <Bus0> estará em <Fátima>. Bus<1> é o mais próximo. Está em <Lisboa> a <130 km>**

Algoritmia e Programação

Trabalho Prático (2023-2024)

NOTA: Durante a realização deste trabalho poderão surgir novos requisitos. Desta forma, poderão ser requeridas funcionalidades adicionais. Por exemplo, os dados iniciais podem ser obtidos de um ficheiro de texto.

Normas:

- O trabalho deverá ser realizado em grupos de dois alunos. A formação dos grupos tem de ser comunicada ao docente das aulas PL, até ao final da 8ª semana de APROG;
- O trabalho deve ser submetido, por todos os alunos, no Moodle até às 23:30 horas do dia 2 de dezembro de 2023. A partir da data indicada, a nota do trabalho será penalizada 20% por cada dia de atraso e não se aceitam trabalhos após dois dias das datas indicadas;
- Após a entrega, nas aulas práticas seguintes, cada grupo terá de defender o trabalho submetido, perante o professor, para avaliação;
- A submissão no moodle deve ser um ficheiro ZIP contendo toda a estrutura do projeto e ficheiros necessários ao seu funcionamento. O nome do ficheiro deve obedecer à seguinte norma: **"APROG_LEI_<turma>_<nºaluno1>_<nºaluno2>.zip";**

  Exemplo: "APROG_LEI_DA_11223344_55667788.zip"
- A não defesa do trabalho implica a não avaliação do mesmo.

Na medida do possível, o trabalho deve ser realizado de forma equitativa pelos elementos do grupo. Nesse sentido, sugere-se a seguinte distribuição das funcionalidades pedidas:

```
ALUNO1: a) c) e) g) i)
ALUNO2: b) d) f) h) j)
```

Critérios de avaliação:

Trabalho de grupo

- Funcionalidades 50%
- Modularização 30%
- Estruturas de dados 10%
- Organização do código 10%

Desempenho individual 100%

*Nota final individual = Desempenho individual * Trabalho de grupo*