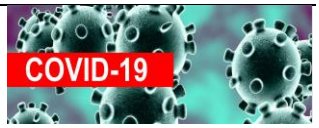

Enunciado do Projeto

Importação, Normalização e Classificação de Dados

1. Objetivo do Projeto

Pretende-se desenvolver um programa em C para extrair informação útil de um ficheiro com dados sobre a incidência do Covid-19 num determinado país.

O programa consiste num interpretador de comandos que o utilizador usa para obter diversos tipos de informação, principalmente informação estatística.

1.1 Representação dos dados em Memória

Cada paciente é representado, obrigatoriamente, pela estrutura de dados `Patient` apresentada na Figura 1, sendo `Date` um tipo de dados apropriado para guardar uma data.

Cada região é representada, obrigatoriamente, pela estrutura de dados `Region` apresentada no abaixo.

```
typedef struct date {
   unsigned int day, month, year;
} Date;

typedef struct patient{
    long int id;
    char sex[6]; // {“male”, “female”}
    int birthYear;
    char country[40]; //birth country
    char region[40];
    char infectionReason[100];
    long int infectedBy; // id of the infected patient
    Date confirmedDate;
    Date releasedDate;
    Date deceasedDate;
    char status[10]; // {“isolated, “released”, “deceased”}
} Patient;

typedef struct region{
    char name[40];
    char capital[40];
    int population;
    floatarea;
    } Region;
```

Figura 1 – Tipos de dados

Nota : Considere-se que todo o paciente em isolamento está infetado.

Na implementação dos comandos descritos neste enunciado podem definir/utilizar outros tipos de dados auxiliares que achem úteis para a resolução dos problemas.

1.2 Dados de entrada

Existem dois tipos de ficheiro com dados:
- Ficheiro de dados sobre os pacientes;
- Ficheiros de dados sobre as regiões.

Ambos os ficheiros se encontram em formato CVS. E a primeira linha dos ficheiros é uma linha com os cabeçalhos e não contem dados.

Ficheiro dos pacientes (cada linha corresponde a informação sobre um paciente)

```
<id>;<sex>;<birth_year>;<birthcountry>;<region>;<infection_reason>;<infected_by>;<confirmed_date>;
<released_date>;<deceased_date>;<state>
...
```

Ficheiro com os dados das regiões (cada linha corresponde a informação sobre uma região)

```
<name>;<capital>;<area>;<population>
...
```

O valor <*confirm_date*> <released_date> e <deceased_date>encontra-se no formato "dd/mm/aaaa".

Pode-se assumir que não existem ficheiros "mal-formados".

A. No caso do ficheiro de dados sobre pacientes podem existir campos em branco.
   - No caso de um campo tipo data vir em branco, a data deve ser inicializada a 0/0/0
   - No caso da birth_year ou infectedBy vir a branco devem ser inicializados a -1.
   - No caso de um campo do tipo string vir em branco deve ser inicializado com uma string "" de comprimento 0.
   - Exemplo de linhas possíveis para o ficheiro dos pacientes:

```
1000000001;male;1964;Korea;Seoul;overseas inflow;;23/01/2020;05/02/2020;;released

7000000005;female;;Korea;Jeju-do;overseas inflow;;24/03/2020;;;isolated
```

B. No caso do ficheiro dos dados das regiões
   - Todos os campos devem ter valores.

Juntamente com este enunciado são disponibilizados 2 ficheiros de entrada para testes:

- patients.csv
- regions.csv

Após a divulgação do enunciado será disponibilizado no Moodle um exemplo com **alguns** dos resultados esperados na execução da aplicação para estes ficheiros.

1.3 Utilização de TADs

É obrigatória a manutenção em memória da informação importada:

- dos pacientes exclusivamente numa instância do ADT List, sendo ListElem o tipo Patient (definido em 1.1)
- das regiões exclusivamente numa instância de ADT Map, sendo ValueElem do tipo Region (definido em 1.1) e o KeyElem de um tipo apropriado que permita guardar uma string;

Não é permitido alterar as interfaces lecionadas dos TAD, nomeadamente os ficheiros list.h e map.h. Estas instâncias serão designadas doravante por "coleções".

1.4 Comandos

Há exatamente 14comandos que o programa deve implementar, que serão apresentados de seguida; 2 comandos para carregamento de dados, 10 comandos para mostrar resultado de cálculos sobre os dados, 1 comando para sair da aplicação e 1 comando para limpeza dos dados em memória.

Notas:

- Cada comando é representado por uma palavra que pode ser escrita pelo utilizador em maiúsculas ou em minúsculas, não importa.
- Sempre que um comando necessitar de algum input, e.g., Id de um paciente, este deve ser solicitado ao utilizador.
- Sempre que um comando necessitar de informação que não está carregada, o comando deve indicar que informação está em falta, i.e., "No patient data available..." e/ou "No region data available...".

A forma exata como os resultados devem ser mostrados no ecrã será descrita em seguida.

A. Os comandos base são os seguintes:

✓ **LOADP**
- Pede o nome dum ficheiro de pacientes, abre o ficheiro e carrega-o em memória (ver Secção 1.2), mostrando o número de doentes importadas. Os restantes comandos passarão a atuar sobre o novo conteúdo da coleção. Se o ficheiro não puder ser aberto, escreve `File not found` e a coleção fica vazia.

✓ **LOADR**
- Abre o ficheiro "regions.csv" e carrega-o em memória (ver 1.2), mostrando o número de regiões importadas. Se o ficheiro não puder ser aberto, escreve `File not found` e a coleção respetiva fica vazia.

✓ **CLEAR**
- Limpa a informação atualmente em memória. Deverá indicar o número de registos que foram descartados, e.g., `"<N> records deleted from <Patients | Regions>"`

✓ **QUIT**
- Sai do programa, libertando toda a memória alocada para as coleções.

B. Os comandos de indicadores simples (os cálculos requeridos só precisam de processar informação da coleção dos pacientes) são os seguintes:

✓ **AVERAGE**

- Mostra as seguintes médias:

```
Average Age for deceased patients: <avg1>
Average Age for released patients: <avg2>
Average Age for isolated patients: <avg3>
```

Para cálculo destas médias só deve ter-se em conta os pacientes com data de nascimento conhecida.

✓ **FOLLOW**

- Dado um id de um paciente mostra a sequência de contaminação: No seguinte formato:

```
Following Patient: ID:<ID>, SEX: <sex>, AGE: <age>, COUNTRY/REGION:
<country> / <Region>, STATE: <state>
contaminated by Patient: ID:<ID>, SEX: <sex>, AGE: <age>, COUNTRY/REGION:
<country> / <Region>, STATE: <state>
contaminated by Patient: ID:<ID>, SEX: <sex>, AGE: <age>, COUNTRY/REGION:
<country> / <Region>, STATE: <state>
```

....

Caso não exista informação sobre quem contaminou o paciente

- o campo está vazio (ou)
- id dado não é encontrado

```
Patient: ID:<ID>, SEX:<sex>, AGE: <age>, COUNTRY/REGION: <country> /
<Region>, STATE: <state>
contaminated by:   unknown
```

✓ **SEX**

- Mostra a percentagem de pacientes:
  - do sexo feminino
  - do sexo masculino
  - de sexo desconhecido (não existe informação nos dados)

```
Percentage of Females: <value>%
Percentage of Males: <value>%
Percentage of unknown: <value>%
Total of patients: <value>
```

✓ **SHOW**

Mostra os dados de um determinado paciente dado o seu id.

```
ID:<ID>
SEX: <sex>
AGE: <age>
COUNTRY/REGION: <country> / <Region>
INFECTION REASON: <reason>
STATE: <state>
NUMBER OF DAYS WITH ILLNESS: <value>
```

Caso não se consiga determinar a idade ou o número de dias de doença deve aparecer `unknown.`

Nota: para o caso de pessoas com no estado isolado o cálculo deve ser realizado com a data mais recente de contaminação encontrada nos registos.

✓ **TOP5**

- Mostra de forma decrescente os 5 pacientes que demoraram mais tempo a recuperar. Cada paciente deve ser mostrado no seguinte formato:

```
ID:<ID>
SEX: <sex>
AGE: <age>
COUNTRY/REGION: <country> / <Region>
INFECTION REASON: <reason>
STATE: <state>
NUMBER OF DAYS WITH ILLNESS: <value>
```

Os pacientes que não se consegue determinar o número de dias de doença não devem ser considerados no cálculo.

Caso não se consiga determinar a idade deve aparecer *unknown*.

Caso haja empate deve-se optar pelo paciente mais velho

✓ **OLDEST**

- Mostra a lista dos pacientes mais idosos de cada sexo, de acordo com o seu ano de nascimento. Para cada paciente deve ser mostrado a informação no seguinte formato:

```
FEMALES:
1 - ID:<ID>, SEX: <sex>, AGE: <age>,COUNTRY/REGION: <country> / <Region>, STATE: <state>
2-
…
MALES:
1 - ID:<ID>, SEX: <sex>, AGE: <age>,COUNTRY/REGION: <country> / <Region>, STATE: <state>
2-
…
```

Os pacientes que não se consegue determinar a idade não devem ser considerados no cálculo.

✓ **GROWTH**

- Dado uma data <date>, mostra a taxa de crescimento do número de mortes e de infetados relativamente ao dia anterior.
  Nota: considera-se que um doente quando é isolado é porque está infetado e que os calculos são referentes ao numero de novos casos em cada dia e não aos valores acumulados.
  As taxas devem ser apresentadas no seguinte formato:

```
Date:<dayBefore(date)>
Number of dead: <number_of_deads>:
Number of isolated: <number_of_isolated>

Date:<date>
Number of dead: <number_of_deads>:
Number of isolated: <number_of_isolated>
```

`Rate of new infected: <rate1>`
`Rate of new dead: <rate1>`

Caso não exista nehum registo para a data introduzida deve apresentar a mensagem:
`There is no record for day <date>`
Nota: taxa de crescimento=(presente-passado)/passado

✓ **MATRIX**
- Cria uma matriz 6x3 de inteiros com informação sobre o número total de pessoas isoladas falecidas e curadas por faixa etária tal como se ilustra na Figura 2.

Nota: Os valores da matriz, apresentados na figura, são meramente ilustrativos.

	Isolated	Deceased	Released
[0-15]	200	0	100
[16-30]	300	2	60
[31-45]	349	22	83
[46-60]	451	25	98
[61-75]	400	28	90
[76...[	501	50	101

Figura 2 – Matrix

C. Os comandos de indicadores complexos (os cálculos requeridos precisam dos dados da coleção de pacientes e da coleção das regiões) são os seguintes:

✓ **REGIONS**
- Mostra a lista de regiões por ordem alfabética, que tem pessoas ainda doentes. A mesma é ordenada alfabeticamente.

✓ **REPORT**
- Cria um ficheiro com o nome `report.txt`, onde é mostrado a taxa de mortalidade e a taxa de incidência, total e por região.
  No ecrã mostra: `Report created` caso o ficheiro tenha sido criado com sucesso e `Report not created` caso contrário. Nota: se não houver dados da população total de um dada região, deverá aparece: `unknown (no population data)`. Os resultados devem aparecer no seguinte formato:

```
<country_ name> Mortality:<value>%   Incident Rate: <value>%      Lethality: <value>%

<region> Mortality: <value>%         Incident Rate : <value>%     Lethality: <value>%

<region> Mortality: <value>%         Incident Rate : <value>%     Lethality: <value>%

<region> unknown (no population data)
```

Lethality (%) = deaths / cases x 100
Mortality (% per 10.000 inhabitants):deaths / population x 10.000
Incident rate(%): infected/population x 100

2 Relatório e Documentação

2.1 Documentação

Todo o código deve ser documentado utilizando a **documentação Doxygen**.

A mesma deve ser gerada para formato HTML e entregue a respetiva pasta "html" junto com o projeto.

2.2 Relatório

No relatório deverão constar as seguintes secções (para além de capa com identificação dos alunos e índice):

a) Descrição breve dos ADTs utilizados, qual o tipo de implementação utilizada e porquê (comparação de eficiências para o problema de aplicação).
b) Para cada comando (exceto CLEAR, e QUIT) fornecer:
    - A complexidade algorítmica da respetiva implementação, tendo em conta as complexidades algorítmicas das funções dos ADTs utilizadas (dependem da implementação escolhida).
c) Escolha de 3 funcionalidades do tipo B e C, onde apresentam o algoritmo implementado em pseudo-código;
d) Limitações: Quais os comandos que apresentam problemas ou não foram implementados;
e) Conclusões: Análise crítica do trabalho desenvolvido.

3 Tabela de Cotações e Penalizações

A avaliação do trabalho será feita de acordo com os seguintes princípios:

- **Estruturação:** o programa deve estar estruturado de uma forma modular e procedimental;
- **Correção:** o programa deve executar as funcionalidades, tal como pedido.
- **Legibilidade e documentação:** o código deve ser escrito, formatado e comentado de acordo com o standard de programação definido para a disciplina.
- **Desempenho:** Os algoritmos implementados devem ter em conta a complexidade do mesmo, valorizando-se a implementação de algoritmos com menor complexidade. A gestão da memória deverá ser feita corretamente, garantindo que a mesma é libertada quando não está a ser utilizada. Utilização da ferramenta Valgrind, para validar a correta gestão de memória.

A nota final obtida, cuja tabela de cotações se apresenta a seguir, será ponderada de acordo com os princípios acima descritos.

Descrição	Cotação (valores)
Leitura de comandos, tratamento de situação de ficheiro inexistente/vazio , limpeza de memória e saída do programa (QUIT)	2
Importação de dados (comandos LOAD)	1,5
Comandos AVERAGE	1
Comando FOLLOW	1,5
Comando SEX	1
Comando SHOW	1
Comando TOP5	1,5
Comando OLDEST	1
Comando GROWTH	1,5
Comando MATRIX	2
Comando REGIONS	1
Comando REPORT	2
Relatório e Documentação	3
TOTAL	20

A seguinte tabela contém penalizações a aplicar:

Descrição	Penalização
Uso de variáveis globais	até 2
Não separação de funcionalidades em funções/módulos	até 3
Não libertação de memória	até 3
Não comentar o programa	até 1
Não utilização dos ADTs obrigatórios	Anulado

4 Instruções e Regras Finais

O IDE a utilizar fica ao critério dos alunos, mas, caso não utilizem o IDE usado na disciplina (i.e., VS-Code), terão que, **antes de submeter, criar os respetivos projetos finais no IDE VS-Code**

O não cumprimento das regras a seguir descritas implica uma penalização na nota do trabalho prático. Se ocorrer alguma situação não prevista nas regras a seguir expostas, essa ocorrência deverá ser comunicada ao respetivo docente de laboratório de ATAD.

Regras:

a) O Projeto deverá ser elaborado por **dois alunos do mesmo docente de laboratório**.

b) A nota do Projeto será atribuída individualmente a cada um dos elementos do grupo após a discussão. As discussões poderão ser orais e/ou com perguntas escritas. As orais poderão ser feitas com todos os elementos do grupo presentes em simultâneo ou individualmente. E poderão ser feitas remotamente via plataforma zoom.

c) A apresentação de relatórios ou implementações plagiadas leva à imediata atribuição de nota zero a todos os trabalhos com semelhanças, quer tenham sido o original ou a cópia.

d) No rosto do relatório e nos ficheiros de implementação deverá constar o número, nome e turma dos autores e o nome do docente a que se destina.

e) O trabalho deverá ser submetido no moodle, no link do respetivo docente de laboratórios criado para o efeito, até às **11:00 do dia 30 de Junho**. Para tal terão que criar uma pasta com o nome: **nomeAluno1_númeroAluno1-nomeAluno2_númeroAluno2**, onde colocarão o ficheiro do relatório em formato **pdf** e uma pasta com o projeto VS Code (pasta com os respetivos ficheiros) da implementação das aplicações a desenvolver. Os alunos terão de submeter essa **pasta compactada em formato ZIP**. Apenas será permitido submeter um ficheiro.

f) Não serão aceites trabalhos entregues que não cumpram na íntegra o ponto anterior.

g) As datas das discussões serão publicadas após a entrega dos trabalhos.

(fim de enunciado)