Programação – Trabalho Prático

Época Especial de Setembro

Licenciatura em Engenharia Informática: 1º ano - 2º semestre 2015/2016

Gestão de uma Clínica Médica

Notas prévias:

- O enunciado é propositadamente vago, genérico e incompleto em alguns pontos. O que se pretende é que os alunos avaliem as opções existentes e escolham a que considerarem mais apropriada para cada uma das situações com que se depararem. Todas as escolhas devem ser referidas e justificadas no relatório.
- O programa entregue deve ter uma interface simples e amigável, indicando o que pode ser feito em cada situação. Não são valorizados programas com interfaces gráficos.
- Deve distribuir o código fonte por vários ficheiros (no mínimo, dois ficheiros com código) e utilizar *header files* para estabelecer a relação entre eles.

1. Introdução

Pretende-se que desenvolva um programa que efetue a gestão da informação de uma clínica médica (médicos e pacientes) e que processe todas as operações relacionadas com a marcação de consultas.

2. Descrição Detalhada do Programa

2.1. Gestão da informação de médicos e pacientes (30%)

Existem duas entidades que o programa tem que manipular: médicos e pacientes. Para cada médico que trabalhe na clínica é necessário considerar a seguinte informação:

- Nome;
- Especialidade[1];
- Horário de trabalho: hora de entrada e de saída;

Por sua vez, cada paciente da clínica tem associada a seguinte informação:

- Nome
- Idade;
- Histórico das 3 últimas consultas. O histórico de cada consulta deve conter a seguinte informação: Tipo de consulta (normal/urgente), data e médico.

[1] Ao implementar o programa, pode assumir que a especialidade tem apenas uma palavra.

A informação sobre médicos e pacientes está armazenada em 2 ficheiros de texto. O formato genérico dos ficheiros deverá ser o seguinte.

Ficheiro de Médicos (exemplo para 3 médicos):

```
Joao Silva
Neurologia 9.30 - 17.00
Ana Maria Santos
Pediatria  10.00 - 19.00
Sandra Almeida
Dermatologia    14.00 - 17.45
```

Ficheiro de Pacientes: Exemplo para 2 pacientes em que o primeiro já tem 2 consultas registadas e o segundo ainda não tem nenhuma consulta.

```
Arnaldo Antunes
56
2 consultas
Normal - 13/08/2016 - Joao Silva
Normal - 17/05/2016 - Joao Silva
Ana Sofia Ramos
23
0 consultas
```

Nota: os ficheiros não têm no início a indicação de quantos médicos ou pacientes estão armazenados.

No início da execução do programa, toda a informação sobre médicos e pacientes é passada para vetores dinâmicos de estruturas e as operações são efetuadas diretamente nestas variáveis. O programa deve permitir executar as seguintes operações:

- Listagem completa de médicos e pacientes;
- Pesquisar médicos por especialidade;
- Listagem de pacientes que tenham tido consultas de uma determinada especialidade;
- Listagem de pacientes que tenham tido consultas num determinado período;
- Outras listagens/consultas que considere relevantes.

Durante a execução, não é permitido adicionar/alterar/eliminar médicos ou pacientes. Caso sejam necessárias, estas operações são feitas diretamente no ficheiro de texto quando o programa não está a ser executado, nem foi interrompido o período de marcação diária (ver ponto 2.3).

2.2. Marcação de Consultas (40%)

Os pacientes da clínica podem solicitar a marcação de uma consulta para o dia atual[2]. O programa deve fazer esta gestão automaticamente, de modo a associar um médico e um período horário a cada pedido feito. A agenda contendo informação relativa à gestão das consultas descrita neste ponto deve ser armazenada numa estrutura dinâmica do tipo lista ligada (lista simples, lista de listas, vetor de listas ou outra estrutura com características semelhantes). A agenda dinâmica completa deve armazenar informação para toda a clínica, mas, por uma questão de organização, poderá criar sub-agendas individuais para cada médico. Nesta agenda individual, cada médico tem uma lista das suas marcações para o dia corrente, indicando em cada uma delas qual o paciente, qual a hora de início e qual a hora final.

Relembra-se que todo o agendamento diz respeito ao dia atual. O programa não contempla a marcação de consultas para os dias seguintes. Fica ao critério do aluno encontrar a melhor organização possível para esta estrutura dinâmica.

As regras para efetuar uma marcação solicitada por um paciente são as seguintes:

- A consulta é pedida para uma determinada especialidade. O paciente pode ou não indicar um médico específico da especialidade em causa.
- A marcação da consulta pode ser feita em 2 modos:
    - Urgente: a marcação é feita o mais cedo possível;
    - Normal: neste caso, o paciente indica qual o período preferencial para a marcação da consulta. Pode ser referido um intervalo horário (por exemplo, entre as 14.00 e as 17.00) ou um período (por exemplo, no período da tarde). Fica ao critério do aluno encontrar a melhor maneira de solicitar informação sobre o período em que deve ser agendada a consulta.
- A duração das consultas varia de acordo com a faixa etária do paciente. Fica ao critério dos alunos escolher como armazenar esta informação. Pode, por exemplo, estar guardada numa variável local (um vetor ou uma estrutura) da função que faz a marcação de consultas.
- Um paciente pode marcar apenas 1 consulta por dia.
- O programa, quando confrontado com o pedido de marcação, deve procurar na agenda do dia qual o período em que a consulta pode ser realizada. Poderá fazer a pesquisa na agenda individual de um médico específico (caso o paciente tenha referido o seu nome) ou então procurar em todas as agendas individuais dos médicos dessa especialidade (caso o paciente, só tenha referido a especialidade desejada). Ao efetuar a procura deve ter em atenção as seguintes restrições:
    - O pedido do paciente deve ser levado em consideração (consulta urgente: marcar o mais depressa possível; consulta normal: tentar marcar para o período solicitado);
    - Nenhum médico pode atender mais do que um paciente ao mesmo tempo;
    - Os médicos têm um horário de trabalho que deve ser satisfeito;
    - Todos os médicos fazem uma pausa de 5 minutos entre consultas.

[2] Consulte no final do enunciado uma maneira simples de obter a data atual.

O programa deve permitir efetuar as seguintes funcionalidades:

- Marcação de uma nova consulta. Esta opção recebe e valida os dados de um novo pedido de consulta. Caso seja possível efetua a marcação, atualizando a agenda dinâmica;
- Listagem completa da agenda: mostra toda a informação relativa às marcações desse dia;
- Eliminar uma consulta: uma das consultas marcada é eliminada e removida da agenda.

2.3. Interromper a Marcação (20%)

Durante o período de marcações diário, o programa pode ser interrompido. Neste caso o programa termina, mas, antes disso, deve gravar toda a estrutura dinâmica de marcação de consultas num ficheiro binário.
Quando o programa recomeçar, deve existir uma opção para ir buscar a informação ao ficheiro binário, reconstruir a agenda dinâmica e recomeçar o processo de marcação. Durante a interrupção, os ficheiros de texto não devem ser modificados para não corromper a informação.

2.4. Final do dia (10%)

No final da execução diária, o ficheiro de pacientes deve ser atualizado com informação das consultas realizadas. Pode assumir que todas as consultas que estejam inseridas na estrutura dinâmica foram realizadas. Imediatamente antes de terminar, o programa acede ao ficheiro de texto e atualiza a informação das consultas realizadas. Caso um paciente já tenha informação sobre 3 consultas no ficheiro e tenha feito uma nova consulta nesse dia, a referência mais antiga deve ser eliminada, sendo substituída pela nova.
Neste modo de finalização do programa (final do dia de trabalho), não é necessário guardar nenhum backup das consultas. Todas as consultas foram realizadas e no próximo dia as marcações começarão com uma agenda vazia.

3. Normas para a realização do trabalho prático

O trabalho deve ser realizado **individualmente**.

Data limite de entrega do trabalho prático: **23:55** do dia **7 de setembro de 2016**. Pode ser entregue 1 dia depois (penalização de 25%) ou 2 dias depois (penalização de 50%)

Material a entregar:

- **Até às 23.55 do dia 07/09/2016:** entregar via *moodle* um ficheiro compactado em formato **ZIP**, contendo o relatório, o código fonte comentado e os ficheiros de dados necessários para o funcionamento do programa.
  O nome do ficheiro ZIP deve obrigatoriamente ter o seguinte formato: ***Prog_pX_NumAluno.zip***, em que X é a identificação da turma prática que frequenta.

- **No início da defesa:** entregar, ao professor responsável pela defesa, uma cópia impressa do relatório. O conteúdo do documento deve ser igual ao submetido via *moodle*.

Os trabalhos serão sujeitos a **defesa obrigatória** em data a anunciar. As defesas incluirão uma explicação detalhada do código e implementação de alterações.

Relatório

Deve ser entregue um relatório contemplando os seguintes pontos:

- Apresentação das principais estruturas de dados, justificando as escolhas feitas;
- Apresentação detalhada das estruturas dinâmicas implementadas;
- Descrição dos ficheiros utilizados;
- Justificação para as opções tomadas em termos de implementação;
- Estrutura geral do programa, incluindo a apresentação de algoritmos de alto nível que clarifiquem as suas principais funcionalidades;
- Pequeno manual de utilização.

Avaliação

A cotação do trabalho é de **8 valores** e a sua entrega substitui a nota prática (testes laboratoriais e trabalho) obtida na época normal de avaliação.

Cada uma das componentes a implementar tem o peso (em %) definido no ponto 2. Para esta componente da avaliação não foi definida nenhuma nota mínima.

A deteção de plágio parcial ou total implica a anulação imediata de todos os trabalhos envolvidos.

Critérios de Avaliação:

- Definição das estruturas de dados;
- Funcionalidades implementadas;
- Correção do código implementado, com especial ênfase na manipulação de estruturas dinâmicas e manipulação de ficheiros;
- Qualidades das soluções algorítmicas encontradas para a resolução dos problemas;
- Simplicidade/funcionalidade do interface com o utilizador;
- Estruturação e documentação do código fonte;
- Relatório;
- Defesa.

Como obter a data num programa em C?

```
#include <time.h>
#include <stdlib.h>
#include <stdio.h>

int main()
{
    int dia, mes, ano;
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);

    dia = tm.tm_mday;
    mes = tm.tm_mon + 1;
    ano = tm.tm_year + 1900;

    printf("Data: %d-%d-%d\n", dia, mes, ano);
    return 0;
}
```