

Programação – Trabalho Prático

Licenciatura em Engenharia Informática: 1º ano - 2º semestre

2016/2017

Zoo Vale das Flores

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 deve ser implementado em C standard, i.e., não deve ter instruções que o tornem específico para um determinado ambiente/plataforma de desenvolvimento. Preferencialmente deverá ser seguida a norma C99.
- 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 comunicação entre eles. Todos os ficheiros devem ter a identificação do aluno (nome completo e número), em comentário, à cabeça.

1. Introdução

Pretende-se que desenvolva um programa em linguagem C que permita gerir a informação dos animais que pertencem a um jardim zoológico. Em concreto, deverá ser possível conhecer as características básicas dos animais, a sua localização e as suas relações familiares. Ao longo do enunciado são descritas as principais funcionalidades a implementar.

2. Animais e Áreas do Zoo

Os 2 principais tipos de informação a manipular pelo programa são os seguintes:

2.1. Áreas do Zoo

O zoo está dividido num conjunto de espaços, que podem ser uma jaula ou um espaço vedado. Cada uma destas zonas contém alguns animais. As áreas são caracterizadas por um identificador alfanumérico único (por exemplo, o nome da área) e por uma capacidade (peso total dos animais que comporta). Pode assumir que o identificador é uma palavra única sem espaços. Cada área tem ainda indicação das suas fronteiras, i.e., dos espaços adjacentes. A configuração do zoo garante que um dado espaço tem, no máximo, 3 outros locais adjacentes.

2.2. Animais

Cada animal é definido pelo seguinte conjunto de características:

- Espécie: Identificador único, comum a todos os animais da mesma espécie. Deve ser uma sequência alfanumérica sem espaços;
- Número de Série: Identificador único do animal dentro da espécie a que pertence. Pode ser numérico ou alfanumérico e deve ser uma sequência sem espaços;
- Nome;
- Peso;
- Localização: cada animal está localizado numa das áreas existentes no zoo;
- Família: cada animal tem indicação de quem são os seus pais (caso se encontrem no zoo) e possui também uma lista dos seus filhos que se encontram no zoo.

3. Programa a Implementar

3.1 Gestão das Áreas do Zoo

A descrição completa das várias áreas do zoo é armazenada num **ficheiro de texto**, em que cada linha tem informação completa sobre uma área: identificador, capacidade e áreas adjacentes. Na figura seguinte ilustra-se um ficheiro para um zoo com 4 áreas:

<i>AreaA</i>	500	2	<i>AreaB</i>	<i>AreaC</i>
<i>AreaB</i>	200	1	<i>AreaA</i>	
<i>AreaC</i>	100	2	<i>AreaA</i>	<i>AreaD</i>
<i>AreaD</i>	1000	1	<i>AreaC</i>	

Neste exemplo, a primeira área tem identificador *AreaA*, 500 kg de capacidade e 2 áreas adjacentes (*AreaB* e *AreaC*).

No início da execução, toda a informação sobre as áreas deve ser transferida para um **vetor de estruturas dinâmico**. Se este ficheiro não existir, o programa deve terminar imediatamente. Durante a execução apenas deve ser usada/manipulada a informação que se encontra no vetor. No final da execução, a informação do vetor deve ser usada para atualizar o ficheiro de texto.

Há duas operações que podem mudar a distribuição das áreas durante a execução:

- Criar uma nova área: o utilizador indica as características da nova área, incluindo as áreas já existentes que vão ficar adjacentes. O programa deve garantir que nenhuma área fique com mais do que 3 adjacências. No processo de criação, a nova área não tem animais.
- Eliminar uma área existente: esta operação só é válida se nenhum animal se encontrar nesta área nessa altura.

3.2 Animais

Todos os animais existentes no zoo pertencem a uma determinada área, não podendo existir animais sem localização atribuída. Quando o programa não está a ser executado, a informação completa sobre os animais encontra-se armazenada num **ficheiro binário**. No início da execução, esta informação deve ser transferida para memória e mantida numa estrutura dinâmica que tenha o formato genérico de **lista ligada**. A implementação pode recorrer a uma ou várias listas ligadas, simples ou com outra organização. No final da execução, a informação no ficheiro deve ser atualizada. Caso o ficheiro não exista no início da execução, o programa deve assumir que o zoo ainda não tem animais.

A leitura do ficheiro binário de animais pode originar vários problemas de incompatibilidade de informação entre animais e locais que devem ser resolvidos de forma simples pelo programa. Exemplos de potenciais problemas são:

- O animal tem indicação de pertencer a uma zona que não existe no zoo: neste caso, o animal pode ser simplesmente ignorado;
- Os animais especificados para pertencer a uma determinada área ultrapassam a sua capacidade. O programa pode ignorar os animais que provocam o ultrapassar do limite ou tentar a redistribuição para uma área que ainda tenha capacidade disponível.

O programa deve implementar soluções igualmente simples para outros problemas que eventualmente possam surgir.

O programa deve permitir efetuar as seguintes operações sobre os animais:

1. Listagem da informação:
 - a. Listagem completa de todos os animais do zoo;
 - b. Listagem dos animais que se encontram numa determinada localização.
 - c. Listagem de todos os animais de uma determinada espécie.
 - d. Listagem completa da informação relativa a um animal, incluindo a identificação dos seus ascendentes e descendentes diretos (caso existam)
2. Eliminar Animais: um animal pode ser retirado do zoo, devendo desaparecer toda a informação relativa a ele (incluindo as relações familiares com outros animais que se mantenham no zoo).
3. Transferir animal: um animal pode ser transferido para uma outra localização dentro do zoo, desde que as áreas envolvidas sejam adjacentes e a capacidade do novo local não seja ultrapassada.
4. Nascimento de um animal: um descendente pode ser criado a partir de 1 ou 2 animais que já existam no zoo. Caso estejam 2 progenitores envolvidos no nascimento, deve garantir-se que são da mesma espécie e que se encontram no mesmo local. O nome da cria é indicado pelo utilizador e o programa deve gerar automaticamente um novo identificador único dentro da espécie a que pertence. No processo de nascimento, 20% do peso de cada progenitor é transferido para o descendente.

5. Adicionar animais: à exceção do nascimento descrito no ponto anterior, a única possibilidade para adicionar animais ao zoo é via ficheiro de texto. Nesta funcionalidade o utilizador indica o nome de um ficheiro de texto e o programa vai aí buscar informação sobre os animais a adicionar ao zoo. O **ficheiro de texto** deve ter o seguinte formato genérico, em que a informação de cada animal surge numa única linha (espécie, nome, peso e área onde deve ser colocado):

<i>Leao</i>	<i>Simba</i>	<i>250</i>	<i>AreaA</i>
<i>Zebra</i>	<i>Leopoldo</i>	<i>100</i>	<i>AreaB</i>

Ao processar este ficheiro, o programa deve atribuir automaticamente o identificador único do animal dentro da espécie a que pertence. As espécies indicadas no ficheiro podem já existir no zoo ou pode ser o primeiro animal dessa espécie a entrar. Se a área indicada não existir ou se a colocação do animal ultrapassar a capacidade do local, o programa deve gerar uma mensagem de erro a explicar o que se passa e ignorar esse animal no ficheiro. Os animais adicionados via ficheiro de texto não têm família no zoo.

3.3 Funcionalidade Extra: Incompatibilidade entre Animais

Na versão base, o zoo não considera incompatibilidades entre animais e permite a coexistência no mesmo espaço de, por exemplo, leões e gazelas. Obviamente esta é uma questão importante que deveria ser considerada na distribuição dos animais pelos espaços.

Nesta funcionalidade extra pretende-se que proponha, explique e implemente um mecanismo que assegure a separação de animais incompatíveis dentro do zoo. Caso implemente esta funcionalidade extra, o programa que entregar deve permitir gerir o zoo das duas formas (a incompatibilidade entre animais deve poder estar ativa ou inativa).

4. Normas para a realização do trabalho prático

O trabalho deve ser **realizado individualmente**. O trabalho só pode ser entregue uma vez e numa única data. A nota obtida é válida em todas as épocas de avaliação do ano letivo 2016/2017.

Data limite de entrega do trabalho prático: 23:55 do dia 11 de Junho de 2017. 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 11/06/2017:** 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. Caso tenham usado o Netbeans para a implementação, deverão incluir no ZIP todos os ficheiros do respetivo projeto.

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*.

Defesa:

Os trabalhos serão sujeitos a **defesa obrigatória** em data a anunciar. As defesas poderão incluir:

- i) Demonstração do funcionamento do programa;
- ii) Explicação detalhada do código;
- iii) Implementação de alterações / novas funcionalidades.

As defesas serão feitas nos computadores dos laboratórios, utilizando o Netbeans em ambiente Windows. Antes da submissão deverão certificar-se que o trabalho submetido está pronto para correr neste tipo de ambiente.

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 base é de **6 valores** e corresponde aos pontos 3.1 e 3.2 do enunciado. A funcionalidade extra, descrita no ponto 3.3, tem uma cotação máxima adicional de 10% (0.6 valores).

Esta componente da avaliação não tem 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;
- Simplicidade/funcionalidade do interface com o utilizador;
- Estruturação e documentação do código fonte;
- Relatório;
- Defesa.