

Estruturas de Dados e Algoritmos

Fundamentais

(ano letivo 2015-16)

e-fólio A

Este enunciado constitui o elemento de avaliação designado por “e-fólio A” no âmbito da avaliação contínua e tem a cotação total de 4 valores. A sua resolução deve ser entregue até às 23h55 do dia 14 de abril pelos alunos que escolheram a modalidade de avaliação contínua.

A resolução deve ser entregue através de um único ficheiro compactado .zip, que:

- (i) contém os ficheiros .cpp que constituem o código dos programas, prontos a serem compilados;
- (ii) contém um ficheiro pdf de formato livre, com um relatório sucinto com informações complementares de modo a permitir uma fácil compreensão do trabalho realizado. É desnecessário incluir uma listagem integral do código.
- (iii) O nome do ficheiro .zip a entregar deve seguir a seguinte convenção para o seu nome,

“NumeroAluno-PrimeiroNome-Apelido-21046-efA.zip”

Por exemplo, um aluno com número 327555 e nome Paulo ... Costa, deverá dar o seguinte nome ao ficheiro, “327555-Paulo-Costa-21046-efA.zip”

O ficheiro deve ser única e exclusivamente entregue através do recurso “E-fólio A” disponibilizado na plataforma (Nota: apenas é visível para os alunos inscritos em avaliação contínua), não sendo aceites trabalhos enviados por outras vias, como por exemplo por e-mail.

Esta é uma prova de avaliação **individual** e não “um trabalho de grupo”. A sua resolução deve provir unicamente do conhecimento adquirido e trabalho original desenvolvido pelo próprio aluno. Os alunos deverão saber distinguir claramente entre discutir os conteúdos abordados na unidade curricular (permitido) e discutir a resolução específica do e-fólio (não permitido).

I

1. Pretende-se desenvolver um programa em linguagem C++ padrão de nome "gdlli.cpp" que aceite comandos para a gestão de uma lista duplamente ligada (double linked list) para armazenar elementos ou itens que são inteiros não negativos. Os comandos de um modo geral devem permitir inserir, remover, alterar, procurar itens na lista além de outros comandos mais específicos.

Os comandos podem ser dados um de cada vez terminados por <ENTER> ou colocados em série num ficheiro com um comando por linha e executados na totalidade utilizando o operador '<' de redirecionamento da entrada padrão na linha de comandos do sistema operativo. Do ponto de vista do programa não deve existir qualquer diferença entre os dois modos de funcionamento.

a)[0.5] Projete as estruturas de dados (classes) adequadas ao programa que se pretende desenvolver. Defina apenas atributos (variáveis membro). Os métodos (funções membro) serão solicitados noutras alíneas. Justifique a presença/necessidade de cada atributo que definir.

Para as alíneas seguintes apenas necessita de apresentar um programa, conforme as funcionalidades que implementou, dado que cada alínea acrescenta funcionalidade à anterior.

Para a execução de cada comando deve indicar e explicar no relatório os métodos (funções membro) que criou/utilizou.

Os métodos e os comandos devem ser implementados tendo em vista a sua eficiência.

b)[2] Projete e implemente uma versão do programa de nome "gdllib.cpp" que implemente os comandos a seguir descritos.

ins0 item

"ins0": comando que significa inserir item no início da lista.

item: um nº inteiro não negativo.

insend item

"insend": comando que significa inserir item no fim da lista.

del0

"del0": comando que significa remover nó do início da lista.

delend

"delend": comando que significa remover nó do fim da lista.

dellist

"dellist": comando que significa remover todos os nós da lista.

Dica: ao fazer as necessárias atribuições a apontadores, atribua primeiro os que se encontram indefinidos, depois os nulos e os restantes.

prt0

"prt0": comando que significa imprimir item do início da lista (uma linha com o item ou com um * se lista vazia).

prtend

"prtend": comando que significa imprimir item do fim da lista (uma linha com o item ou com um * se lista vazia).

prt

"prt": comando que significa imprimir toda a lista (uma linha com os itens separados por um espaço ou um * se lista vazia).

dim

"dim": comando que significa imprimir o nº total de itens na lista.

exit

"exit": comando que termina o programa.

c)[0.75] Partindo da alínea anterior projete e implemente uma versão do programa de nome "gdllc.cpp" que implemente também os comandos a seguir descritos.

find item

"find": comando que significa procurar primeira ocorrência do item na lista e imprimir a sua posição (* se não for encontrado).

del item

"del": comando que significa remover primeira ocorrência do item na lista e imprimir a sua posição (* se não for encontrado).

ins pos item

"ins": comando que significa inserir item na lista.

pos: um nº inteiro não negativo que indica a posição de inserção na lista. Se a posição não for válida deve ignorar o comando e imprimir mensagem de erro.

d)[1] Partindo da alínea anterior projete e implemente a versão integral do programa de nome "gdlli.cpp" que implemente também os comandos a seguir descritos.

sort

"sort": comando que significa ordenar os itens da lista. A estratégia a seguir é criar uma lista auxiliar, procurar e remover sucessivamente o máximo item da lista e inseri-lo na lista auxiliar. No final a lista auxiliar substitui a lista original, ficando a existir apenas uma lista. Indique em notação Big-O a complexidade da execução do comando.

- O programa deve estar identificado com um cabeçalho similar ao seguinte,

```
/*  
** UC: 21045-Estruturas de Dados e Algoritmos Fundamentais  
** e-fólio A 2015-16 (gdlli.cpp)  
**  
** Aluno: 327555 - Paulo Costa  
*/
```

- A inclusão de “print screens” da linha de comandos no relatório só deve ser feita com fundo branco.

- No desenvolvimento do programa em C++ não deve ser utilizada a STL no que respeita às estruturas de dados e algoritmos estudados, devendo o aluno escrever o próprio código. Restrições aplicam-se nomeadamente aos includes <array> <deque> <forward_list> <list> <map> <queue> <set> <stack> <unordered_map> <unordered_set> <vector> e em parte de <algorithm>. Em caso de dúvida questionar o seu uso. Não existem restrições para <string>.

Critérios de correção:

- Programa não compila ou produz avisos (warnings) com `g++ -Wall` => 0 valores.
- Código do programa não está correta e uniformemente indentado de modo a permitir a sua leitura fácil => 0 valores
- Programa não está comentado => 0 valores. Os comentários no programa elucidam questões relevantes do código locais ao comentário.
- Funcionalidade do programa de acordo com o pedido, estrutura, nível de simplicidade e qualidade do código (até 65%)
- Relatório. Explique o como e porquê relativamente às opções e soluções técnicas que tomou para a estrutura e funcionamento do programa (até 35%)

Nota ética: Nunca é de mais referir que o código a apresentar como solução para este e-fólio deve ser 100% original do aluno. A probabilidade de duas pessoas que efetivamente não comunicaram entre si, apresentarem programas “quase iguais” é considerada nula. Isto é válido para qualquer par de alunos (cópia), assim como entre um aluno e qualquer outra pessoa, em particular através da Internet (cópia/plágio), onde existem inúmeras soluções e código para os mais variados problemas, em sites, fóruns, blogs, etc.

Cumpra estritamente as normas de realização individual, como se estivesse num exame com consulta, onde pode consultar a documentação mas não pode falar com ninguém.

FIM