

---

# Estruturas de Dados e Algoritmos

## Fundamentais

(ano letivo 2016-17)

### 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 que constituem o código dos programas, prontos a serem compilados;
- (ii) contém um ficheiro “relatorio.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,

“NúmeroAluno-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).

É permitido **partilhar** nos fóruns dados de entrada/saída do programa solicitado.

# I

1. Pretende-se desenvolver um programa em linguagem C++ padrão de nome "gcslli.cpp" que aceite comandos para a gestão de uma lista **circular** simplesmente ligada (circular single 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.

O programa pode funcionar em dois modos distintos:

(i) Modo interativo, em que funciona como um interpretador de comandos. Neste modo o programa executa permanentemente um ciclo em que imprime a prompt “\$”, espera que o utilizador digite um comando terminado por <ENTER>, executa o comando e volta a imprimir a prompt. O programa termina quando o utilizador introduz o comando “exit”. O programa funciona em modo interativo quando a nível do sistema operativo é invocado com o argumento “cmd”. Exemplo:

```
>> gcslli cmd
```

(ii) Modo não interativo, em que o programa lê comandos da entrada padrão, com um comando por linha. Neste modo o programa não imprime nenhuma prompt e cada comando é executado imediatamente antes de ser lido o comando da linha seguinte. O programa termina quando for encontrado o comando “exit” ou não houver mais comandos para executar. Exemplo:

```
>> gcslli < ficheiro-com-comandos.txt
```

Em ambos os modos o programa deve ignorar linhas em branco.

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)[2.25] Projete e implemente uma versão do programa que implemente os comandos a seguir descritos.

## **ins0 item**

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

**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 n.º total de itens na lista (uma linha com o formato dim=NNN).

**c)[0.75]** Extenda o programa anterior de modo a implementar também os comandos a seguir descritos.

**find item**

"find": comando que significa procurar primeira ocorrência do item na lista e imprimir a sua posição (uma linha com o formato find pos=NNN ou um \* se não encontrado).

**del item**

"del": comando que significa remover primeira ocorrência do item na lista. (imprimir uma linha com um \* se não 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)[0.5]** Extenda o programa anterior de modo a implementar também o comando a seguir descrito.

**inv**

"inv": comando que significa inverter a ordem dos itens da lista. A estratégia a seguir é criar uma lista auxiliar, remover e inserir sucessivamente cada item da lista 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: 21046-Estruturas de Dados e Algoritmos Fundamentais
** e-fólio A 2016-17 (gcslli.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 com o compilador gcc ou g++ => 0 valores.
- Programa não contém os respetivos #include para cada função ou classe que utiliza => 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.
- Programa não cumpre as especificações de entrada de dados => 0 valores na componente de funcionalidade, não será possível testar o programa.
- Programa não cumpre as especificações de saída de dados, não gerando os dados solicitados ou gerando dados a mais não solicitados => 0 valores na componente de funcionalidade, não será possível testar o programa. (sugestão: utilize uma flag em todos os prints que efetuar de dados não solicitados, entregue o programa com flag=0)
- O relatório deve dar uma boa ideia do trabalho efetuado sem ser necessário ver o código. Explique o como e porquê relativamente às opções e soluções técnicas que tomou para a estrutura e funcionamento do programa (até 35%).
- Funcionalidade do programa de acordo com o pedido (até 35%).
- Estrutura, nível de simplicidade e qualidade do código (até 30%).

**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.

Cumpre 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