UNIVERSIDADE ABERTA
www.uab.pt

Código 21045

# Estruturas de Dados e Algoritmos Avançados

(ano letivo 2017-18)

## 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 30 de outubro 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 .c ou .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.** Escreva um programa em linguagem C ou C++ padrão, de nome `cbcm`, que cifre e decifre mensagens segundo um algoritmo de cifra por encadeamento de blocos (Cipher Block Chaining Mode) com blocos de 8 bits, recorrendo à associação de caixas P (8 bits) e S (4 bits) em cascata PSP, de modo a formar um dispositivo que implementa uma cifra de produto.

- O programa deve ter como entrada de dados a entrada padrão, como saída de dados a saída padrão e aceitar obrigatoriamente como primeiro argumento a letra c (para cifrar) ou d (para decifrar).

- Opcionalmente como segundo argumento, pode aceitar uma chave K que será utilizada para as caixas tipo P. A chave da caixa S é obtida acrescentando K+8 (somar 8 a cada dígito da chave) à chave K, obtendo-se assim uma chave hexadecimal de 16 dígitos. Sendo a chave opcional, no caso de não ser fornecida uma pelo utilizador deve ser usada internamente por defeito a chave K="10537642".

- O programa deve validar os argumentos recebidos em termos do caracter de comando e comprimento da chave dada.

- A cifra de produto é obtida com uma cascata PSP, onde P são caixas tipo P iguais de 8 bits e S contém 2 caixas tipo S iguais de 4 bits aplicadas respetivamente aos bits 0-3 e 4-7.

- A mensagem deve ser lida da entrada padrão, processada e escrita na saída padrão em blocos de 8 bits (1 byte) de cada vez. De modo algum a entrada deve ser toda previamente lida para memória.

- No desenvolvimento do programa não devem ser usadas funções matemáticas reais, nomeadamente cálculo de potências e logaritmos.

- Para valor inicial (IV) deve ser utilizado IV=0xA5.

- Para decifrar a mensagem o programa deve calcular automaticamente as chaves inversas, ou seja, é sempre dado ao programa a chave usada para cifrar.

- Admitindo ">> " como prompt para a linha de comandos do sistema operativo, são exemplos de utilização do programa utilizando redirecionamento da entrada/saída padrão a nível da linha de comandos do sistema operativo com '<', '>' e '|', (se necessário efetue uma revisão deste tópico),

- Para cifrar um ficheiro de nome "texto.txt" com a chave "76543210", e guardar o resultado num ficheiro de nome "texto.cif",

```
>> cbcm c 76543210  <texto.txt  >texto.cif
```

- Para decifrar um ficheiro de nome "texto.cif", que foi cifrado com a chave "76543210", e guardar o resultado num ficheiro de nome "texto.txt",

```
>> cbcm d 76543210  <texto.cif  >texto.txt
```

- Para cifrar e decifrar num só passo, utilizando a chave por defeito, útil para verificar o bom funcionamento do programa,

```
>> cbcm c <texto.txt | cbcm d >texto2.txt
```

onde se deve obter o ficheiro "texto2.txt" igual ao ficheiro "texto.txt"

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

```
/*
** UC: 21045-Estruturas de Dados e Algoritmos Avançados
** e-fólio A 2017-18 (cbcm)
**
** Aluno: 327555 - Paulo Costa
*/
```

- No desenvolvimento do programa em C++ não deve ser utilizada a STL. 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>.

Nota: Se vai desenvolver o seu programa em ambiente Windows, tenha em atenção que (para funcionar e testar em Windows) deve reabrir a entrada e saída padrão para modo binário para que não ocorra a tradução virtual entre "\r\n" e "\n". Em linguagem C pode fazê-lo com a função freopen() de <stdio.h>. Em linguagem C++ pode também fazê-lo com a função freopen() de <cstdio>. Em ambiente Linux ou MAC OSX esta questão não se põe. Os programas serão testados em Linux.

**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, estrutura, nível de simplicidade e qualidade do código (até 65%)

**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