

Programação I

1º Semestre 2019/2020

2º Trabalho

DATA DE ENTREGA: 24 DE Novembro DE 2019

Objectivos: Neste trabalho, os alunos adquirem prática de utilização de instruções de ciclo, *arrays* e funções. Pretende-se que os alunos desenvolvam uma abordagem estruturada (*top-down*) na construção de programas, através da especificação de funções, e organizando-os em módulos (múltiplos ficheiros fonte) quando necessário.

Cada grupo entregará no *site* da sua turma um documento (com extensão docx, doc ou pdf) que descreva a solução encontrada para cada exercício, assim como os ficheiros fonte das respetivas soluções, devidamente comentados, comprimidos num ficheiro (com extensão zip, rar, gz, etc.).

É valorizada a simplicidade das soluções que cumpram o enunciado.

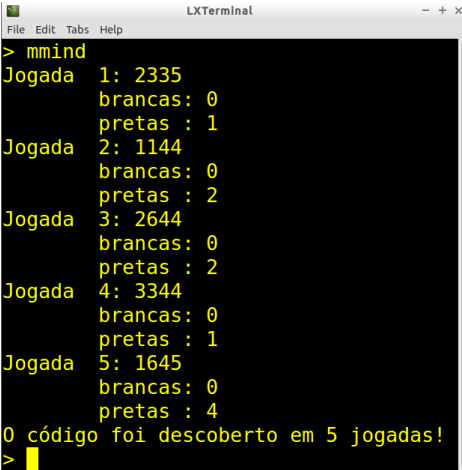
1. Pretende-se disponibilizar um conjunto de funções para formatar nomes (de pessoas) com diferentes critérios, por exemplo convertendo para maiúsculas, retirando espaços a mais, etc. Em anexo são disponibilizados ficheiros com as assinaturas das funções a realizar e comentários que descrevem o comportamento pretendido para cada função (*nomes.h*), o ficheiro com as definições das funções (*nomes.c*, a completar) e um ficheiro com código de teste (*nomes_testes.c*) para verificar a correção das vossas soluções.

Nota: assumo que não são utilizados caracteres acentuados na construção dos nomes.

- a) Complete as funções presentes em *nomes.c* de modo a que todos os testes sejam bem sucedidos.
- b) Acrescente a função de formatação de nomes *name_middle_compressed* apresentada abaixo. Acrescente os testes ao ficheiro *nomes_testes.c* (baseando-se nos testes existentes) para testar esta nova função.

```
/*
 * Descrição:
 *   Gera o formato alternativo para o nome recebido por parâmetro em que apenas
 *   o primeiro nome e o ultimo apelido são apresentados por extenso, estando os nomes
 *   intermédios apresentado na forma de abreviatura.
 *   Espaços iniciais e finais e espaços intermédios a mais devem ser eliminados
 *   Ex: se o nome original for: " pedro manuel vieira rodrigues "
 *       o nome resultante da conversão é: "Pedro M. V. Rodrigues".
 * Parâmetros:
 *   "orig": nome a converter
 *   "result": o nome no formato requerido
 * Retorno:
 *   "true" se o nome for passível de conversão
 *   (tem de ter pelo menos um nome próprio e um apelido).
 *   "false" caso contrário.
 */
bool name_middle_compressed(char orig[], char result[])
```

2. O código em anexo (na pasta `mmind`) tem uma versão incompleta do conhecido jogo Master Mind, onde o jogador humano tenta descobrir o código gerado aleatoriamente pelo computador. O código a descobrir tem quatro cores, dentro de 6 cores disponíveis, representadas por algarismos de 1 a 6, podendo ocorrer cores repetidas. Por cada jogada o jogador humano introduz o possível código através de um número com 4 dígitos (de 1 a 6). O computador apresenta o resultado da jogada indicando o número de cores presentes no código e no lugar certo (pretas) e o número de cores presentes no código mas em diferente posição (brancas). Quando o utilizador descobrir as 4 cores nos seus respetivos lugares o programa termina indicando o número de tentativas usadas.



```
> mmind
Jogada 1: 2335
brancas: 0
pretas : 1
Jogada 2: 1144
brancas: 0
pretas : 2
Jogada 3: 2644
brancas: 0
pretas : 2
Jogada 4: 3344
brancas: 0
pretas : 1
Jogada 5: 1645
brancas: 0
pretas : 4
0 código foi descoberto em 5 jogadas!
>
```

O código está distribuído pelos seguintes ficheiros:

- `mmind.c` – contém o código de apresentação e de execução das jogadas (completo).
- `model.c` – contém as funções que avaliam a jogada e geram a chave secreta aleatoriamente. Este módulo está por implementar.
- `model.h` – contém as declarações das funções presentes em `model.c` (completo).
- `utils.c` - funções utilitárias, por exemplo para a geração de valores aleatórios (completo).
- `utils.h` – declaração das funções definidas em `utils.c` (completo).

- a) Analise cuidadosamente o código disponibilizado e complete as funções presentes no ficheiro `model.c` de modo a ter um programa que permita jogar o Master Mind de acordo a descrição anterior.
- b) Faça as alterações/adições ao código disponibilizado para poder jogar também com a restrição de não poderem existir cores repetidas. O programa deve começar por pedir o modo em que se pretende jogar (com ou sem a possibilidade de cores repetidas).

Bom trabalho

Jorge Martins
Manuel Carvalho
Mário Pinheiro
Nuno Oliveira