



**UNIDADE CURRICULAR:** Introdução à Programação

**CÓDIGO:** 21173

**DOCENTE:** José Coelho

**A preencher pelo estudante**

**NOME:** Jorge Rafael Teixeira Pires

**N.º DE ESTUDANTE:** 2300122

**CURSO:** Licenciatura de Engenharia Informática

Autoavaliação:

Critérios	Alínea A	Alínea B	Alínea C	Alínea D
Funcionalidade	0.5	0.5	0.5	0.5
Qualidade	0.8			
Relatório dimensão / forma	0.19			
Relatório conteúdo	0.19			
Relatório testes	0.13			

Autoavaliação de acordo com os critérios de correção: +0.1 na nota do e-fólio  
Ver critérios de correção no espaço da UC ([aqui](#))

## TRABALHO / RESOLUÇÃO:

### Alínea A – Terminada

Estrutura de dados:

- MAX\_SIZE 100
  - Constante máxima para inicialização inicial
- int countLines, countChars;
  - São contadores de linhas e caracteres, são usadas no Main() e na ProcessLine();

Funções:

- char\* AllocateLine()
  - Aloca memória para uma linha de caracteres, utiliza MAX\_SIZE
- void ProcessLine(char \*line, int \*countLines, int \*countChars);
  - Esta função recebe uma linha de texto como entrada e atualiza a contagem de linhas e caracteres.

Funcionamento global:

- A função main() começa por definir as variáveis necessárias, trabalhar a entrada de dados e imprimir os resultados.

Notas:

- Casos de teste e resultado dos testes de qualidade em anexo

## **Alínea B – Terminada**

Estrutura de dados:

- `int isFirstWhiteLine, lastLineIsWhite;`
  - Definem estados, para controlo de acções.
- `int countLines, countChars;`
  - São contadores de linhas e caracteres, são usadas no `Main()` e na `ProcessLine()`

Funções:

- `void ProcessLine(char *line, int *countLines, int *countChars);`
  - Esta função sofreu uma alteração da Alínea A, atualizando contagens e estados.

Funcionamento global:

- A função `main()` começa por definir as variáveis necessárias, trabalhar a entrada de dados e estados e no fim imprime os resultados.

Notas:

- Casos de teste e resultado dos testes de qualidade em anexo

## Alínea C – Terminada

Estrutura de dados:

- `struct Player`
  - Estrutura que define os valores necessários para um player, neste caso com `*name`, `lastGameScore`, `score`, `wins`, servindo para nome, ultima pontuação, pontuação total e jogos ganhos.
- `int readingPlayers, lastLineIsWhite, NumberOfPlayers`
  - Definem estados, para controlo de ações.
- `int *line`
  - Ponteiro para armazenar temporariamente linha do ficheiro;
- `Player *players`
  - Ponteiro para armazenar os jogadores

Funções:

- `void HandleEmptyLine(int *readingPlayers, int *lastLineIsWhite)`
  - Manipula uma linha vazia no ficheiro.
- `void CreatePlayer(char *line, Player **players, int *NumberOfPlayers)`
  - Esta função recebe uma linha de entrada e cria uma estrutura de jogador com base nas informações da linha. O jogador criado é adicionado à matriz de jogadores e o número de jogadores é atualizado conforme necessário.
- `void UpdatePlayer(char *line, int *readingPlayers, int *lastLineIsWhite, Player *players, int *NumberOfPlayers)`
  - Atualiza as informações do jogador com base na linha fornecida.
- `void HandlePlayer(char *line, Player *players, int *NumberOfPlayers)`
  - Manipula um jogador com base na linha de entrada fornecida.
- `*GetName(char *line)`
  - Esta função recebe uma linha como entrada e retorna o nome extraído dela.
- `int GetPoints()`
  - Recupera os pontos
- `void FreePlayers(Player **players, int NumberOfPlayers)`

- Esta função libera a memória alocada para uma matriz de ponteiros de Jogador.
- `void PrintResult(Player *players, int NumberOfPlayers)`
  - Esta função recebe uma matriz de estruturas de Jogador e o número de jogadores como entrada, ela imprime o resultado de cada jogador
- `.void Updatewinners(Player *players, int NumberOfPlayers)`
  - Atualiza os vencedores com base nas pontuações dos jogadores..

Funcionamento global:

- A função `main()` começa por definir as variáveis necessárias, trabalhar a entrada de dados separando por lógicas e estados e no fim imprime os resultados.

Notas:

- Casos de teste e resultado dos testes de qualidade em anexo

## **Alínea D – Terminada**

Funções:

- `int ComparePlayers(const void *a, const void *b)`
  - A função `ComparePlayers` compara os jogadores baseado no seu valor de forma a ser utilizada em conjunto com o `qsort`.

Funcionamento global:

- A função `main()` começa por definir as variáveis necessárias, trabalhar a entrada de dados separando por lógicas e estados e no fim imprime os resultados.

Notas:

- Casos de teste e resultado dos testes de qualidade em anexo

## **Considerações gerais**

Nos testes de qualidade todas as alíneas acusaram problemas que desconsidereei, por acreditar que sejam falsos positivos, por causa das variáveis estarem em inglês.

## ANEXOS

### Testes

*Tabela 1 - Casos de teste*

<p>Liam Foster Samantha Miller Victoria Brooks Ava Mitchell Benjamin Carter</p> <p>Samantha Miller: 23 Benjamin Carter: 41 Victoria Brooks: 12 Liam Foster: 35 Ava Mitchell: 7</p> <p>Liam Foster: 14 Ava Mitchell: 27 Samantha Miller: 5 Victoria Brooks: 49 Benjamin Carter: 38</p> <p>Ava Mitchell: 19 Samantha Miller: 42 Liam Foster: 8 Victoria Brooks: 31 Benjamin Carter: 13</p>	<p>Oliver Smith Emily Williams James Brown Sophia Davis William Turner Charlotte White Alexander Johnson</p> <p>Sophia Davis:35 William Turner:63 Charlotte White:21</p> <p>Sophia Davis:94 William Turner:13</p> <p>Sophia Davis:91 William Turner:82 Charlotte White:14 Alexander Johnson:82</p>
<p><b>Saída Alínea A:</b> 5 68</p>	<p><b>Saída Alínea A:</b> 7 95</p>
<p><b>Saída Alínea B:</b> 3 15</p>	<p><b>Saída Alínea B:</b> 2 9</p>
<p><b>Saída Alínea C:</b> Liam Foster - 57 pontos, 0 vitórias. Samantha Miller - 70 pontos, 1 vitórias. Victoria Brooks - 92 pontos, 1 vitórias. Ava Mitchell - 53 pontos, 0 vitórias. Benjamin Carter - 92 pontos, 1 vitórias.</p>	<p><b>Saída Alínea C:</b> Oliver Smith - 0 pontos, 0 vitórias. Emily Williams - 0 pontos, 0 vitórias. James Brown - 0 pontos, 0 vitórias. Sophia Davis - 220 pontos, 2 vitórias. William Turner - 158 pontos, 1 vitórias. Charlotte White - 35 pontos, 0 vitórias. Alexander Johnson - 82 pontos, 0 vitórias.</p>
<p><b>Saída Alínea D:</b> 1. Benjamin Carter - 92 pontos, 1 vitórias. 2. Victoria Brooks - 92 pontos, 1 vitórias. 3. Samantha Miller - 70 pontos, 1 vitórias. 4. Liam Foster - 57 pontos, 0 vitórias. 5. Ava Mitchell - 53 pontos, 0 vitórias.</p>	<p><b>Saída Alínea D:</b> 1. Sophia Davis - 220 pontos, 2 vitórias. 2. William Turner - 158 pontos, 1 vitórias. 3. Alexander Johnson - 82 pontos, 0 vitórias. 4. Charlotte White - 35 pontos, 0 vitórias. 5. Emily Williams - 0 pontos, 0 vitórias. 6. James Brown - 0 pontos, 0 vitórias. 7. Oliver Smith - 0 pontos, 0 vitórias.</p>

```
● PS D:\UAb - LEI\Introdução à Programação\efolioB> tcc -run .\2300122A.c
5 68
○ PS D:\UAb - LEI\Introdução à Programação\efolioB> █
```

Figura 1 - Exemplo 1, Alínea A

```
● PS D:\UAb - LEI\Introdução à Programação\efolioB> tcc -run .\2300122A.c
7 95
○ PS D:\UAb - LEI\Introdução à Programação\efolioB> █
●
```

Figura 2 - Exemplo 2, Alínea A

```
● PS D:\UAb - LEI\Introdução à Programação\efolioB> tcc -run .\2300122B.c
3 15
○ PS D:\UAb - LEI\Introdução à Programação\efolioB> █
```

Figura 3 - Exemplo 1. Alínea B

```
● PS D:\UAb - LEI\Introdução à Programação\efolioB> tcc -run .\2300122B.c
2 9
● PS D:\UAb - LEI\Introdução à Programação\efolioB> █
```

Figura 4 - Exemplo 2, Alínea B

```
● PS D:\UAb - LEI\Introdução à Programação\efolioB> tcc -run .\2300122C.c
Liam Foster - 57 pontos, 0 vitórias.
Samantha Miller - 70 pontos, 1 vitórias.
Victoria Brooks - 92 pontos, 1 vitórias.
Ava Mitchell - 53 pontos, 0 vitórias.
Benjamin Carter - 92 pontos, 1 vitórias.
● PS D:\UAb - LEI\Introdução à Programação\efolioB> █
```

Figura 5 - Exemplo 1. Alínea C

```

● PS D:\UAb - LEI\Introdução à Programação\efolioB> tcc -run .\2300122C.c
Oliver Smith - 0 pontos, 0 vitórias.
Emily Williams - 0 pontos, 0 vitórias.
James Brown - 0 pontos, 0 vitórias.
Sophia Davis - 220 pontos, 2 vitórias.
William Turner - 158 pontos, 1 vitórias.
Charlotte White - 35 pontos, 0 vitórias.
Alexander Johnson - 82 pontos, 0 vitórias.
○ PS D:\UAb - LEI\Introdução à Programação\efolioB> 

```

Figura 6 - Exemplo 2, Alínea C

```

● PS D:\UAb - LEI\Introdução à Programação\efolioB> tcc -run .\2300122D.c
1. Benjamin Carter - 92 pontos, 1 vitórias.
2. Victoria Brooks - 92 pontos, 1 vitórias.
3. Samantha Miller - 70 pontos, 1 vitórias.
4. Liam Foster - 57 pontos, 0 vitórias.
5. Ava Mitchell - 53 pontos, 0 vitórias.
● PS D:\UAb - LEI\Introdução à Programação\efolioB> 

```

Figura 7 - Exemplo 1. Alínea D

```

● PS D:\UAb - LEI\Introdução à Programação\efolioB> tcc -run .\2300122D.c
1. Sophia Davis - 220 pontos, 2 vitórias.
2. William Turner - 158 pontos, 1 vitórias.
3. Alexander Johnson - 82 pontos, 0 vitórias.
4. Charlotte White - 35 pontos, 0 vitórias.
5. Emily Williams - 0 pontos, 0 vitórias.
6. James Brown - 0 pontos, 0 vitórias.
7. Oliver Smith - 0 pontos, 0 vitórias.
○ PS D:\UAb - LEI\Introdução à Programação\efolioB> 

```

Figura 8 - Exemplo 1. Alínea D

## Qualidade:

### A.

```

PS D:\UAb - LEI\Introdução à Programação\efolioB> .\qualidade.exe .\2300122A.c
#EQ4: Nomes sem significado
13|   char *line = malloc(MAX_SIZE * sizeof(char));
PS D:\UAb - LEI\Introdução à Programação\efolioB> |

```

Figura 9 - Qualidade alínea A



**B.**

```
PS D:\UAb - LEI\Introdução à Programação\efolioB> .\qualidade.exe .\2300122B.c
#EQ4: Nomes sem significado
13|   char *line = malloc(MAX_SIZE * sizeof(char));
PS D:\UAb - LEI\Introdução à Programação\efolioB> |
```

*Figura 10 - Qualidade alínea B*

**C.**

```
PS D:\UAb - LEI\Introdução à Programação\efolioB> .\qualidade.exe .\2300122C.c
#EQ3: Variáveis com nomes quase iguais (valor: 6)
228|   int readingPlayers = 1, lastLineIsWhite = 0, NumberOfPlayers = 0;
#EQ4: Nomes sem significado
12|   int wins;
PS D:\UAb - LEI\Introdução à Programação\efolioB> |
```

*Figura 11 - Qualidade alínea C*

**D.**

```
PS D:\UAb - LEI\Introdução à Programação\efolioB> .\qualidade.exe .\2300122D.c
#EQ3: Variáveis com nomes quase iguais (valor: 6)
242|   Player *playerB = (Player *)b;
241|   Player *playerA = (Player *)a;
#EQ4: Nomes sem significado
12|   int numberWins;
PS D:\UAb - LEI\Introdução à Programação\efolioB> |
```

*Figura 12 - Qualidade alínea D*