



**Universidade de São Paulo**  
Instituto de Ciências Matemáticas e de Computação  
Departamento de Ciências de Computação  
SCC0201 – Introdução à Ciência da Computação II

## Trabalho 2: Quadro de medalhas

**Professor:** Dr. Rodrigo Fernandes de Mello ([mello@icmc.usp.br](mailto:mello@icmc.usp.br))  
Dr. Moacir Antonelli Ponti ([moacir@icmc.usp.br](mailto:moacir@icmc.usp.br))  
**Estagiário PAE:** Felipe Simões Lage Gomes Duarte ([fgduarte@icmc.usp.br](mailto:fgduarte@icmc.usp.br))  
Gabriel de Barros Paranhos da Costa ([gbpcosta@icmc.usp.br](mailto:gbpcosta@icmc.usp.br))  
Tiago Santana de Nazaré ([tiagosn@usp.br](mailto:tiagosn@usp.br))

### 1 Objetivo do Trabalho

Você deverá implementar um programa que permita ao usuário simular um quadro de medalhas de uma competição esportiva. Esse quadro de medalhas deverá ter diversas funcionalidades que serão descritas a seguir.

### 2 Proposta

Seu programa deverá possibilitar que o usuário realize diversas operações de ordenação e busca em um quadro de medalhas. Ao iniciar, o seu programa deverá receber o nome de um arquivo que contém os dados do quadro de medalhas e carregá-lo. Para facilitar o processo de desenvolvimento e correção, seu programa deverá implementar 5 diferentes módulos: (i) ordenação (ordem crescente), (ii) ordenação (ordem decrescente), (iii) busca por país (nome), (iv) busca por país (nomes em intervalo), (v) busca por quantidade de medalhas (intervalo). Cada um destes módulos deverão ser executados de acordo com uma opção inteira que será fornecida via `stdin`.

Para esse trabalho será necessária a implementação de funções de busca e ordenação. A função de ordenação deve ser criada de forma a receber, como parâmetro, um ponteiro para outra função que execute a comparação entre um par de elementos conforme necessário. Assim, deve-se utilizar **apenas uma função** para realizar a ordenação, alterando a forma de comparação conforme necessário. Importante: seu trabalho será invalidado caso não utilize esse recurso, mesmo que a saída do programa esteja conforme a esperada.

Você está livre para utilizar o método de ordenação que achar mais adequado.

A cada opção escolhida o programa deverá imprimir o resultado da opção escolhida. Quando o resultado for em forma de tabela, cada coluna da mesma deve ocupar um espaço equivalente a 20 caracteres com o texto alinhado à direita. As colunas são separadas por um caractere espaço.

#### 2.1 Quadro de medalhas

O arquivo que contém o quadro de medalhas está organizado de tal forma que as informações de cada país estão em uma linha e cada linha possui quatro colunas (nome, quantidade de medalhas de ouro, quantidade de medalhas de prata e quantidade de medalhas de bronze) separadas por um caractere espaço. A partir dessas informações, uma nova coluna deve ser criada com a quantidade total de medalhas de cada país quando necessário realizar a impressão do quadro. Exemplo de arquivo contendo um quadro de medalhas:

```
COLOMBIA 27 14 31\n
BRASIL 41 40 60\n
EUA 103 81 81\n
CUBA 36 27 34\n
CANADA 78 69 70\n
EQUADOR 7 9 16\n
GUATEMALA 6 1 3\n
MEXICO 22 30 \n
ARGENTINA 15 29 31\n
VENEZUELA 8 22 20\n
PERU 4 4 6\n
CHILE 5 6 18\n
```

## 2.2 Opções

Como citado anteriormente, o sistema deverá possuir comportamentos distintos de acordo com a opção de entrada.

### 2.2.1 Opção 1

O programa deverá imprimir no `stdout` o quadro de medalhas carregado do arquivo de entrada em ordem crescente, sendo que o critério utilizado para realizar a ordenação será fornecido via `stdin`. Ao final de cada linha, uma quebra de linha deverá ser impressa, i.e., um ‘`\n`’.

Os possíveis critérios para a ordenação são:

- **g** – Ordenar por número de medalhas de ouro. Desempates por número de medalhas de prata, número de medalhas de bronze e nome (ordem alfabética), respectivamente.
- **s** – Ordenar por número de medalhas de prata. Desempates por número de medalhas de ouro, número de medalhas de bronze e nome (ordem alfabética), respectivamente.
- **b** – Ordenar por número de medalhas de bronze. Desempates por número de medalhas de ouro, número de medalhas de prata e nome (ordem alfabética), respectivamente.
- **t** – Ordenar por quantidade total de medalhas. Desempates por número de medalhas de ouro, número de medalhas de prata e nome (ordem alfabética), respectivamente.
- **n** – Ordenar por nome (ordem alfabética).

Exemplo (usando o arquivo mostrado anteriormente):

```
1
g
```

Saída:

País	Ouros	Pratas	Bronzes	Total
PERU	4	4	6	14
CHILE	5	6	18	29
GUATEMALA	6	1	3	10
EQUADOR	7	9	16	32
VENEZUELA	8	22	20	50
ARGENTINA	15	29	31	75
MEXICO	22	30	3	55
CUBA	36	27	34	97
COLOMBIA	41	14	31	86
BRASIL	41	40	60	141
CANADA	78	69	70	217
EUA	103	81	81	265

### 2.2.2 Opção 2

Similar a opção 1, porém o quadro deve ser impresso em ordem decrescente.

Exemplo (usando o arquivo mostrado anteriormente):

```
2
b
```

Saída:

País	Ouros	Pratas	Bronzes	Total
EUA	103	81	81	265
CANADA	78	69	70	217
BRASIL	41	40	60	141
CUBA	36	27	34	97
COLOMBIA	41	14	31	86
ARGENTINA	15	29	31	75
VENEZUELA	8	22	20	50
CHILE	5	6	18	29
EQUADOR	7	9	16	32
PERU	4	4	6	14
MEXICO	22	30	3	55
GUATEMALA	6	1	3	10

### 2.2.3 Opção 3

Para essa opção, o programa deve receber como entrada o nome de um país e buscar suas informações no quadro de medalhas. Caso não haja nenhum país com o nome passado, a mensagem “Nao ha um pais chamado %nome\_buscado%.” deve ser impressa.

Exemplo (usando o arquivo mostrado anteriormente):

```
3
BRASIL
```

Saída:

```
Nome:  BRASIL\n
Ouros: 41\n
Pratas: 40\n
Bronzes: 60\n
Total: 141\n
```

### 2.2.4 Opção 4

O programa deve receber como entrada 2 caracteres que delimitam um intervalo e imprimir todos os países cuja inicial de seu nome esteja no intervalo especificado.

Exemplo:

4  
C F

Saída:

País	Ouros	Pratas	Bronzes	Total
COLOMBIA	41	14	31	86
EUA	103	81	81	265
CUBA	36	27	34	97
CANADA	78	69	70	217
EQUADOR	7	9	16	32
CHILE	5	6	18	29

### 2.2.5 Opção 5

O programa deve receber como entrada 2 inteiros que delimitam um intervalo e imprimir todos os países cujo número de medalhas de ouro esteja no intervalo especificado.

Exemplo:

5  
30 70

Saída:

País	Ouros	Pratas	Bronzes	Total
COLOMBIA	41	14	31	86
BRASIL	41	40	60	141
CUBA	36	27	34	97

### 2.2.6 Opção 0

Finaliza o programa.

## 2.3 Exemplo de entrada

```
1.board\n
1\n
g\n
2\n
b\n
3\n
BRASIL\n
4\n
C F\n
5\n
30 70\n
0\n
```

### 3 IMPORTANTE

- Utilize alocação dinâmica (memória heap) para armazenar os dados;
- Não esqueça de liberar toda memória alocada antes de encerrar a execução do seu programa;
- Crie quantas funções achar necessário, a modularização do sistema será levado em consideração no processo de correção.
- **PRAZO FINAL: 17/08 - 23:59:59**