



# Ranking Olímpico

Professora Patrícia Dockhorn Costa, [patricia.d.costa@ufes.br](mailto:patricia.d.costa@ufes.br)

---

## Descrição

Nesta prova você deve implementar um programa que exiba o ranking de medalhas das olimpíadas, utilizando uma **Tabela Hash**.

Em um primeiro momento, o programa deve (vale até 6,0):

- Abrir o arquivo de entrada e ler as informações sobre países e suas medalhas;
- Armazenar cada país na tabela hash, realizando o tratamento de colisão com listas encadeadas;
- Caso o país já esteja inserido na tabela hash, use as informações para atualizar os quantitativos de medalhas;
- Deve-se usar o código de 3 caracteres do país como chave de busca.

Na segunda etapa, deve-se (vale até 4,0, assumindo que a primeira etapa foi implementada):

- Alocar um vetor que tenha como tamanho, o número de países distintos do arquivo de entrada, sem repetições;
- Ordenar o vetor, por ordem crescente de medalhas de ouro, prata e bronze, nessa ordem (para isso, use a *qsort* da biblioteca padrão);
- Abrir o arquivo de saída e escrever os países (e suas medalhas) em ordem crescente de medalhas (ouro, prata e bronze);
- Liberar toda a memória alocada.

Considere o arquivo de entrada a seguir, como exemplo (cada linha: país ouro prata bronze total):

```
BEL 1 0 1 2
HKG 1 0 0 1
FIJ 0 1 0 1
JPN 1 0 1 2
KAZ 1 0 1 2
GER 1 0 0 1
BRA 0 0 0 0
CRC 0 0 0 0
ITA 0 1 2 3
DMA 0 0 0 0
ESA 0 0 0 0
GBR 0 1 1 2
CAN 0 1 0 1
AUS 3 2 0 5
CUB 0 0 0 0
CHN 2 0 1 3
USA 1 2 2 5
FRA 1 2 1 4
KOR 1 1 1 3
DOM 0 0 0 0
MGL 0 1 0 1
TUN 0 1 0 1
ESP 0 0 1 1
HUN 0 0 1 1
ECU 0 0 0 0
RSA 0 0 1 1
SWE 0 0 1 1
GRN 0 0 0 0
BRA 0 0 1 1
AUS 3 2 2 7
USA 1 2 3 6
```

O arquivo de saída deve conter uma lista de países, ordenado pelo ranking. Para o exemplo acima, a seguinte saída é esperada:

```
AUS 3 2 2 7
CHN 2 0 1 3
USA 1 2 3 6
FRA 1 2 1 4
KOR 1 1 1 3
BEL 1 0 1 2
JPN 1 0 1 2
KAZ 1 0 1 2
GER 1 0 0 1
HKG 1 0 0 1
ITA 0 1 2 3
GBR 0 1 1 2
CAN 0 1 0 1
FIJ 0 1 0 1
MGL 0 1 0 1
TUN 0 1 0 1
BRA 0 0 1 1
ESP 0 0 1 1
HUN 0 0 1 1
RSA 0 0 1 1
SWE 0 0 1 1
CRC 0 0 0 0
CUB 0 0 0 0
DOM 0 0 0 0
DMA 0 0 0 0
ESA 0 0 0 0
ECU 0 0 0 0
GRN 0 0 0 0
```

#### Regras importantes:

- 1) Use uma tabela hash para armazenar os dados e permitir a busca pelos dados;
- 2) Utilize tipos de dados estruturados;
- 3) Organize sua prova em tad's (por exemplo, tadhash.c, tadhash.h, tadpais.c, tadpais.h, seunome.c);
- 4) Não utilize variáveis globais;
- 5) Escreva seu **nome** em TODOS os arquivos produzidos;
- 6) Deixe os arquivos de responde no mesmo lugar do enunciado;
- 7) Libere toda a memória alocada (use valgrind).

**BOA PROVA!**