

# Exercícios

**Dica:** Na linha de comando, use

```
java SeuPrograma < arquivo.txt
```

para obter os dados a partir de um arquivo previamente preenchido.

**Problema da mochila fracionária** Você é um aventureiro explorando uma caverna. Ao final da expedição, encontra vários itens de valor, cada um com um peso e um valor associado. Porém, sua mochila só suporta uma quantidade limitada de peso. Felizmente, você pode fracionar os itens, levando apenas uma parte do item e, proporcionalmente, o valor associado.

O objetivo é escolher frações dos itens de modo que o valor total carregado na mochila seja o maior possível sem ultrapassar o limite de peso.

## Entrada

Um número inteiro  $n$  representando a quantidade de itens.

Um número real  $W$  representando a capacidade máxima da mochila.

Em seguida,  $n$  linhas com dois números reais cada: valor e peso de cada item.

## Saída

Um número real com duas casas decimais, representando o valor total máximo que pode ser carregado na mochila.

## Exemplo de entrada

```
3
50
60 10
100 20
120 30
```

## Exemplo de saída

**240.00**

**Explicação:** 10 unidades do primeiro item, totalizando 60 + 20 unidades do segundo item, totalizando 100 + 20 unidades do item 3, totalizando 80.

**Dicas**

Comece calculando o valor unitário de cada item. Por exemplo.

Item 1:  $60/10 = 6$

Item 2:  $100 / 20 = 5$

Item 3:  $120 / 30 = 4$

A partir daí, encontre um critério para escolher cada item de forma gulosa.

**Qual a complexidade computacional de seu algoritmo? Explique.**

### **Referências**

CORMEN, Thomas H. et al. **Introduction to Algorithms**. 3. ed. Cambridge: MIT Press, 2009.

Feofiloff, Paulo. **Anotações sobre Algoritmos: Slides**. São Paulo: Instituto de Matemática e Estatística – USP, [s.d.]. Disponível em: <https://www.ime.usp.br/~pf/livrinho-AA/downloads/AA-SLIDES.pdf>. Acesso em: março de 2025.

KLEINBERG, Jon; TARDOS, Éva. **Algorithm Design**. Boston: Pearson, 2006.