

EXERCÍCIOS DE VECTOR

HORSTMANN, C. Conceitos de Computação com o Essencial em C++ / Cap 9 (online)

P9.1. Escreva uma função `double scalar_product(vector<double> a, vector<double> b)` que calcula o produto escalar de dois vetores.

P9.2. Escreva uma função que calcule a soma alternada de todos os elementos em um vetor. P.ex. vetor 1,4,9,16,9,7,4,9,11 calcula $1 - 4 + 9 - 16 + 9 - 7 + 4 - 9 + 11 = -2$

P9.3. Escreva um procedimento `reverse` que inverte a sequência dos elementos em um vetor.

P9.4. Escreva uma função `vector<int> append(vector<int> a, vector<int> b)` que acrescenta um vetor após outro.

P.ex. vetor 1,4,9,16 e b 9,7,4,9,11 então `append` retorna o vetor 1,4,9,16,9,7,4,9,11

P9.5. Escreva uma função `vector<int> merge(vector<int> a, vector<int> b)` que intercala dois arrays, alternando elementos dos dois arrays. Se um array é mais curto do que o outro, então alterne enquanto você puder e depois acrescente os elementos restantes do array mais longo.

P9.6. Escreva uma função `vector<int> merge_sorted(vector<int> a, vector<int> b)` que intercala dois arrays ordenados, produzindo um novo array ordenado. Mantenha um índice para cada array, indicando quanto dele já foi processado. A cada vez, acrescente o menor elemento não processado de qualquer um dos arrays, então avance o índice.

P.ex., a é 1,4,9,16 e b é 4,7,9,9,11 então `merge_sorted` retorna o array 1,4,4,7,9,9,9,11,16

P9.7. Escreva uma função predicao `bool equals(vector<int> a, vector<int> b)` que verifica se dois vetores têm os mesmos elementos na mesma ordem.

P9.8. Escreva uma função predicao `bool same_set(vector<int> a, vector<int> b)` que verifica se dois vetores têm os mesmos elementos em alguma ordem, ignorando multiplicidades.

P.ex., os dois vetores 1,4,9,16,9,7,4,9,11 e 11,11,7,9,16,4,1 seriam considerados idênticos. Você provavelmente vai precisar de uma ou mais funções auxiliares.

P9.9. Escreva uma função predicao `bool same_elements(vector<int> a, vector<int> b)` que verifica se dois vetores têm os mesmos elementos em alguma ordem, com as mesmas multiplicidades.

P.ex., 1,4,9,16,9,7,4,9,11 e 11,1,4,9,16,9,7,4,9 seriam considerados idênticos, mas 1,4,9,16,9,7,4,9,11,3,5,2 e 11,11,7,9,16,4,1 não seriam. Você provavelmente vai precisar de uma ou mais funções auxiliares.

P9.10. Escreva uma função que remove duplicatas de um vetor.

P. ex., se `remove_duplicates` é chamada com um vetor contendo 1,4,9,16,9,7,4,9,11 então o vetor é mudado para 1,4,9,16,7,11