



Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação

SCC0222 – Laboratório de Introdução à Ciência da Computação I

Exercício: Bubble Sort

Docentes: Bruna C. Rodrigues da Cunha (brunaru@icmc.usp.br)
Leo Sampaio Ferraz Ribeiro (leo.ribeiro@icmc.usp.br)
Rudinei Goularte (rudinei@icmc.usp.br)

Pessoas Monitoras: Pietra Gullo Salgado Chaves
Juan Henriques Passos
Marina Souza Figueiredo
Daniel Jorge Manzano
Bernardo Maia Coelho
Ketlen Victoria Martins de Souza
Fernando Valentim Torres

1 Descrição

O **Bubble Sort** é um algoritmo de ordenação simples baseado na comparação de elementos adjacentes. Ele funciona analisando pares consecutivos em uma lista e trocando suas posições quando estão na ordem incorreta. Esse processo de comparação e troca é repetido iterativamente até que toda a lista esteja completamente ordenada.

Desenvolva um programa em **C** que receba um vetor de inteiros e o exiba em ordem crescente. Para isso, implemente uma função `bubblesort` do tipo `void` que deverá receber como parâmetros um vetor de inteiros e o seu tamanho, e executar a ordenação utilizando o algoritmo de Bubble Sort

Passos do Algoritmo:

1. Percorra a lista comparando cada elemento com o próximo (elemento i com $i+1$).
2. Se o elemento atual for maior que o próximo, troque-os de lugar.
3. Repita esse processo para todos os elementos até que nenhuma troca seja necessária durante uma passagem completa pela lista (indicando que a lista está ordenada).

Ao final, o programa deve imprimir:

- O vetor ordenado, separado por espaços.

2 Instruções Complementares

- Submeta o arquivo `.c` com seu código no <https://runcodes.icmc.usp.br>
- O Run Codes só considerará correta a saída do seu programa se estiver **idêntica** à saída esperada.

3 Exemplos de Entrada e Saída

A seguir são apresentados exemplos de entrada e saída para que você teste seu código enquanto desenvolve o exercício. Este são apenas exemplos ilustrativos, somente uma pequena parte das operações está representada. Enquanto estiver desenvolvendo, elabore novos testes para validar seu código.

Entrada

10 10 9 8 7 6 5 4 3 2 1

Saída

1 2 3 4 5 6 7 8 9 10