

```

#include <stdio.h>

int main() {
    // 1. DECLARAÇÃO E INICIALIZAÇÃO DE VARIÁVEIS

    // Vetor com 10 preços (FLOAT)
    float preco[10] = {15.99, 12.50, 17.15, 9.99, 23.70, 18.90, 10.79, 1.90, 80.50,
13.45};

    // Variáveis de controle
    float c = 0; // Variável temporária para a troca (SWAP)
    int i = 0;   // Contador do Loop Externo (Passagens)
    int j = 0;   // Contador do Loop Interno (Comparações)

    // =====
    // 2. EXIBIÇÃO DO VETOR ORIGINAL
    // =====
    printf("=====\\n");
    printf("          BUBBLE SORT DE PREÇOS\\n");
    printf("=====\\n");
    printf("Vetor Original:\\n");

    for(i = 0; i < 10; i++){
        printf("-> R$ %.2f ", preco[i]);
    }
    printf("\\n\\n");

    // =====
    // 3. ORDENAÇÃO CRESCENTE (Menor para o Maior)
    // =====
    printf("--- ORDENAÇÃO CRESCENTE ---\\n");

    // LOOP EXTERNO (i): Controla o número de passagens (N-1 = 9)
    for (i = 0; i < 9; i++) {

        // LOOP INTERNO (j): Controla as comparações. O alcance diminui a cada 'i'.
        for (j = 0; j < 9 - i; j++) {

            // CONDIÇÃO: Se o atual é MAIOR que o próximo, troque.
            if (preco[j] > preco[j+1]) {

                // LÓGICA DE TROCA (SWAP)
                c = preco[j];
                preco[j] = preco[j+1];
                preco[j+1] = c;
            }
        }
    }

    // EXIBIÇÃO DO VETOR ORDENADO EM CRESCENTE
    for(i = 0; i < 10; i++){
        printf("-> R$ %.2f ", preco[i]);
    }
    printf("\\n\\n");
}

```

```

// =====
// 4. ORDENAÇÃO DECRESCENTE (Maior para o Menor)
// =====
printf("--- ORDENAÇÃO DECRESCENTE ---\n");

// Reutilizando o vetor 'preco', que já está ordenado

// LOOP EXTERNO (i): Repetição das passagens (mesma estrutura)
for (i = 0; i < 9; i++) {

    // LOOP INTERNO (j): Repetição das comparações (mesma estrutura)
    for (j = 0; j < 9 - i; j++) {

        // CONDIÇÃO PARA TROCA (SWAP): Se o atual é MENOR que o próximo, troque.
        // (Se o menor está na frente do maior, é a ordem ERRADA para Decrescente)
        if (preco[j] < preco[j+1]) { // <----- O SINAL INVERTIDO!

            // LÓGICA DE TROCA (SWAP) é exatamente a mesma
            c = preco[j];
            preco[j] = preco[j+1];
            preco[j+1] = c;
        }
    }
}

// EXIBIÇÃO DO VETOR ORDENADO EM DECRESCENTE
for(i = 0; i < 10; i++){
    printf("-> R$ %.2f ", preco[i]);
}
printf("\n");
printf("=====\n");

return 0;
}

```

```
=====
BUBBLE SORT DE PREÇOS
=====
```

Vetor Original:

-> R\$ 15.99 -> R\$ 12.50 -> R\$ 17.15 -> R\$ 9.99 -> R\$ 23.70 -> R\$ 18.90 -> R\$ 10.79 -> R\$ 1.90 -> R\$ 80.50 -> R\$ 13.45

--- ORDENAÇÃO CRESCENTE ---

-> R\$ 1.90 -> R\$ 9.99 -> R\$ 10.79 -> R\$ 12.50 -> R\$ 13.45 -> R\$ 15.99 -> R\$ 17.15 -> R\$ 18.90 -> R\$ 23.70 -> R\$ 80.50

--- ORDENAÇÃO DECRESCENTE ---

-> R\$ 80.50 -> R\$ 23.70 -> R\$ 18.90 -> R\$ 17.15 -> R\$ 15.99 -> R\$ 13.45 -> R\$ 12.50 -> R\$ 10.79 -> R\$ 9.99 -> R\$ 1.90

...Program finished with exit code 0

Press ENTER to exit console.