

Código fonte em formato texto:

```
#include <stdio.h>

// Função que aplica o Bubble Sort
// Parâmetros:
// vetor[] -> vetor de valores
// n -> tamanho do vetor
// crescente -> 1 para crescente, 0 para decrescente
void bubbleSort(float vetor[], int n, int crescente) {
    int i, j;
    float temp;

    // O algoritmo faz várias "passadas" pelo vetor
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            // Verifica a condição de troca
            int condicao = crescente ?
                (vetor[j] > vetor[j + 1]) : // crescente
                (vetor[j] < vetor[j + 1]); // decrescente

            if (condicao) {
                // Troca os valores de posição
                temp = vetor[j];
                vetor[j] = vetor[j + 1];
                vetor[j + 1] = temp;
            }
        }
    }
}
```

```
}
```

```
// Função para imprimir o vetor
```

```
void imprimir(float vetor[], int n) {
```

```
    for (int i = 0; i < n; i++) {
```

```
        printf("R$ %.2f ", vetor[i]);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
int main() {
```

```
    // Vetor com preços de 10 produtos (tema escolhido)
```

```
    float precos[10] = {
```

```
        22.50, 10.00, 7.20, 5.90, 18.30,
```

```
        12.40, 8.70, 15.60, 9.90, 25.00
```

```
};
```

```
int n = 10;
```

```
printf("=====\n");
```

```
printf(" ORDENACAO DE PREÇOS - METODO BOLHA\n");
```

```
printf("=====\n\n");
```

```
// Exibe o vetor original
```

```
printf("Lista Original:\n");
```

```
imprimir(precos, n);
```

```
// Ordenando em ordem crescente
```

```
bubbleSort(precos, n, 1);
```

```
printf("\nOrdenado em ordem crescente:\n");
```

```
imprimir(precos, n);
```

```
// Ordenando em ordem decrescente
```

```

bubbleSort(precos, n, 0);

printf("\nOrdenado em ordem decrescente:\n");

imprimir(precos, n);

printf("\n===== \n");

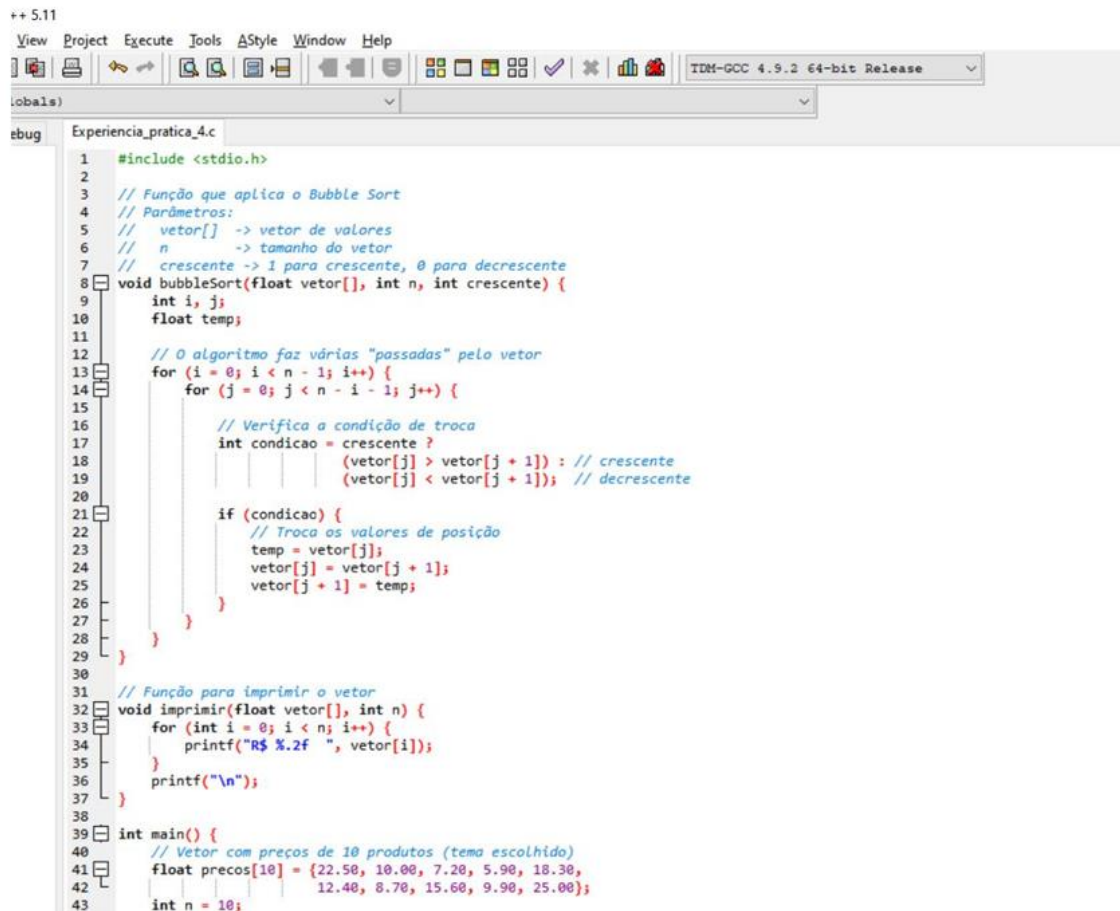
printf(" FIM DA EXECUCAO\n");

printf("===== \n");

return 0;
}

```

Evidências



```

++ 5.11
View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
globals
ebug
Experiencia_pratica_4.c
1 #include <stdio.h>
2
3 // Função que aplica o Bubble Sort
4 // Parâmetros:
5 //   vetor[] -> vetor de valores
6 //   n       -> tamanho do vetor
7 //   crescente -> 1 para crescente, 0 para decrescente
8 void bubbleSort(float vetor[], int n, int crescente) {
9     int i, j;
10    float temp;
11
12    // O algoritmo faz várias "passadas" pelo vetor
13    for (i = 0; i < n - 1; i++) {
14        for (j = 0; j < n - i - 1; j++) {
15
16            // Verifica a condição de troca
17            int condicao = crescente ?
18                (vetor[j] > vetor[j + 1]) : // crescente
19                (vetor[j] < vetor[j + 1]); // decrescente
20
21            if (condicao) {
22                // Troca os valores de posição
23                temp = vetor[j];
24                vetor[j] = vetor[j + 1];
25                vetor[j + 1] = temp;
26            }
27        }
28    }
29 }
30
31 // Função para imprimir o vetor
32 void imprimir(float vetor[], int n) {
33     for (int i = 0; i < n; i++) {
34         printf("R$ %.2f ", vetor[i]);
35     }
36     printf("\n");
37 }
38
39 int main() {
40     // Vetor com preços de 10 produtos (tema escolhido)
41     float precos[10] = {22.50, 10.00, 7.20, 5.90, 18.30,
42                        12.40, 8.70, 15.60, 9.90, 25.00};
43     int n = 10;

```

Project Execute Tools AStyle Window Help

Experiencia_pratica_4.c

```
25     vetor[j + 1] = temp;
26 }
27 }
28 }
29 }
30
31 // Função para imprimir o vetor
32 void imprimir(float vetor[], int n) {
33     for (int i = 0; i < n; i++) {
34         printf("R$ %.2f ", vetor[i]);
35     }
36     printf("\n");
37 }
38
39 int main() {
40     // Vetor com preços de 10 produtos (tema escolhido)
41     float precos[10] = {22.50, 10.00, 7.20, 5.90, 18.30,
42                        12.40, 8.70, 15.60, 9.90, 25.00};
43     int n = 10;
44
45     printf("===== \n");
46     printf("  ORDENACAO DE PREÇOS - METODO BOLHA \n");
47     printf("===== \n \n");
48
49     // Exibe o vetor original
50     printf("Lista Original: \n");
51     imprimir(precos, n);
52
53     // Ordenando em ordem crescente
54     bubbleSort(precos, n, 1);
55     printf("\nOrdenado em ordem crescente: \n");
56     imprimir(precos, n);
57
58     // Ordenando em ordem decrescente
59     bubbleSort(precos, n, 0);
60     printf("\nOrdenado em ordem decrescente: \n");
61     imprimir(precos, n);
62
63     printf("\n===== \n");
64     printf("  FIM DA EXECUCAO \n");
65     printf("===== \n");
66
67     return 0;
68 }
```

ces Compile Log Debug Find Results

Sel: 0 Lines: 67 Length: 2019 Insert Done parsing in 0,141 seconds

pratica_4 - [experiencia pratica 4.dev] - [executing] - Dev-L++ 5.11

arch View Project Execute Tools AStyle Window Help

(globals)

Experiencia_pratica_4.c

```
28 }
29 }
30 }
31 // Função para imprimir o vetor
32 void imprimir(float vetor[], int n) {
33     for (int i = 0; i < n; i++) {
34         printf("R$ %.2f ", vetor[i]);
35     }
36     printf("\n");
37 }
38
39 int main() {
40     // Vetor com preços de 10 produtos (tema escolhido)
41     float precos[10] = {22.50, 10.00, 7.20, 5.90, 18.30,
42                        12.40, 8.70, 15.60, 9.90, 25.00};
43     int n = 10;
44
45     printf("===== \n");
46     printf("  ORDENACAO DE PREÇOS - METODO BOLHA \n");
47     printf("===== \n \n");
48
49     // Exibe o vetor original
50     printf("Lista Original: \n");
51     imprimir(precos, n);
52
53     // Ordenando em ordem crescente
54     bubbleSort(precos, n, 1);
55     printf("\nOrdenado em ordem crescente: \n");
56     imprimir(precos, n);
57
58     // Ordenando em ordem decrescente
59     bubbleSort(precos, n, 0);
60     printf("\nOrdenado em ordem decrescente: \n");
61     imprimir(precos, n);
62
63     printf("\n===== \n");
64     printf("  FIM DA EXECUCAO \n");
65     printf("===== \n");
66
67     return 0;
68 }
```

Resources Compile Log

Processing

- C Compiler

- Command:

Compilation

C:\Users\Jefferson\Desktop\helo world\Debugando\Experiencia_pratica_4.exe

```
=====
ORDENACAO DE PREÇOS - METODO BOLHA
=====

Lista Original:
R$ 22.50 R$ 10.00 R$ 7.20 R$ 5.90 R$ 18.30 R$ 12.40 R$ 8.70 R$ 15.60 R$ 9.90 R$ 25.00

Ordenado em ordem crescente:
R$ 5.90 R$ 7.20 R$ 8.70 R$ 9.90 R$ 10.00 R$ 12.40 R$ 15.60 R$ 18.30 R$ 22.50 R$ 25.00

Ordenado em ordem decrescente:
R$ 25.00 R$ 22.50 R$ 18.30 R$ 15.60 R$ 12.40 R$ 10.00 R$ 9.90 R$ 8.70 R$ 7.20 R$ 5.90

=====
FIM DA EXECUCAO
=====

Process exited after 0.1816 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

