

Entrega IV – Ordenação com Método Bolha em Linguagem C

Aluna: Stephanie Lima da Costa Anicama

Turma: 004

Tema: Tempo de Deslocamento

Objetivo da Atividade

Esta atividade busca desenvolver sua habilidade de organizar dados de forma lógica e eficiente utilizando o método de ordenação Bubble Sort (Método Bolha). Você deverá escolher um dos temas cotidianos a seguir, gerar os dados de forma manual ou simulada, e aplicar a ordenação para apresentá-los de forma crescente e decrescente.

Código

```
#include <stdio.h>
#include <stdlib.h>

// Define uma constante para o tamanho do vetor
#define QTD_ONIBUS 10

int main() {
    int tempos[QTD_ONIBUS];
    int i, j, temp;

    // --- ENTRADA DE DADOS ---
    printf("=====\n");
    printf("  ORDENACAO DE TEMPOS DE ONIBUS (BUBBLE SORT)  \n");
    printf("=====\n");
    printf("Por favor, insira o tempo de chegada (em minutos) de cada onibus:\n\n");

    for (i = 0; i < QTD_ONIBUS; i++) {
        printf("Digite o tempo do onibus %d: ", i + 1);
        scanf("%d", &tempos[i]);
    }

    printf("\n--- Tempos na ordem original ---\n");
    for (i = 0; i < QTD_ONIBUS; i++) {
        printf("%d min | ", tempos[i]);
    }
    printf("\n\n");

    // =====
    // --- LÓGICA DO BUBBLE SORT PARA ORDEM CRESCENTE ---
    // =====
    // O Bubble Sort funciona com dois laços aninhados.
    // O laço externo (com 'i') apenas garante que o processo de comparação
```

```

// se repetirá o suficiente para ordenar todo o vetor.
for (i = 0; i < QTD_ONIBUS - 1; i++) {
    // O laço interno (com 'j') é onde a mágica acontece. Ele percorre o vetor
    // comparando cada elemento com seu vizinho da direita.
    for (j = 0; j < QTD_ONIBUS - i - 1; j++) {

        // CONDIÇÃO: Se o elemento da esquerda for MAIOR que o da direita,
        // eles estão na ordem errada e precisam ser trocados.
        if (tempos[j] > tempos[j + 1]) {

            // INÍCIO DA TROCA (SWAP)
            // 1. Guarda o valor da esquerda (maior) em uma variável temporária.
            temp = tempos[j];
            // 2. Copia o valor da direita (menor) para a posição da esquerda.
            tempos[j] = tempos[j + 1];
            // 3. Pega o valor maior que foi guardado e o coloca na posição da direita.
            tempos[j + 1] = temp;
            // FIM DA TROCA
        }
    }
}

printf("--- Tempos em ORDEM CRESCENTE ---\n");
for (i = 0; i < QTD_ONIBUS; i++) {
    printf("%d min | ", tempos[i]);
}
printf("\n\n");

// =====
// --- LÓGICA DO BUBBLE SORT PARA ORDEM DECRESCENTE ---
// =====
// A lógica é exatamente a mesma, a única coisa que muda é o sinal
// na condição, para inverter a ordem.
for (i = 0; i < QTD_ONIBUS - 1; i++) {
    for (j = 0; j < QTD_ONIBUS - i - 1; j++) {

        // CONDIÇÃO: Se o elemento da esquerda for MENOR que o da direita,
        // eles estão na ordem errada e precisam ser trocados.
        if (tempos[j] < tempos[j + 1]) {

            // A lógica da troca é idêntica à anterior.
            temp = tempos[j];
            tempos[j] = tempos[j + 1];
            tempos[j + 1] = temp;
        }
    }
}

```

```

printf("--- Tempos em ORDEM DECRESCENTE ---\n");
for (i = 0; i < QTD_ONIBUS; i++) {
    printf("%d min | ", tempos[i]);
}
printf("\n\n");

return 0;
}

```

The screenshot shows a C++ IDE with a file named 'Bubble.c'. The code implements a Bubble Sort algorithm. The main function starts with a loop for 'i' from 0 to 'QTD_ONIBUS - 1'. Inside, there's a loop for 'j' from 0 to 'QTD_ONIBUS - i - 1'. A conditional swap is performed if 'tempos[j] < tempos[j + 1]'. The swap logic is: 'temp = tempos[j]; tempos[j] = tempos[j + 1]; tempos[j + 1] = temp;'. The output window shows the program's execution, including the prompt 'Ordemacao de tempos de onibus (Bubble Sort)' and a request for input: 'Por favor, insira o tempo de chegada (em minutos) de cada onibus:'. The user has entered '1' for the first bus. A notification in the bottom right corner indicates 'Compilation successful'.

```

C Bubble.c > main()
7  int main() {
61 // =====
64 // A lógica é exatamente a mesma, a única coisa que muda é o sinal
65 // na condição, para inverter a ordem.
66 for (i = 0; i < QTD_ONIBUS - 1; i++) {
67     for (j = 0; j < QTD_ONIBUS - i - 1; j++) {
68
69         // CONDIÇÃO: Se o elemento da esquerda for MENOR que o da direita,
70         // eles estão na ordem errada e precisam ser trocados.
71         if (tempos[j] < tempos[j + 1]) {
72
73             // A lógica da troca é idêntica à anterior.
74             temp = tempos[j];
75             tempos[j] = tempos[j + 1];
76             tempos[j + 1] = temp;
77         }
78     }
79 }
80
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

Microsoft Windows [versão 10.0.19045.6332]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\rodri\OneDrive\Área de Trabalho\TI CURSO\AULAS\ALGORITIMO\output>cd "c:\Users\rodri\OneDrive\Área de Trabalho\TI CURSO\AULAS\ALGORITIMO\output\output"

c:\Users\rodri\OneDrive\Área de Trabalho\TI CURSO\AULAS\ALGORITIMO\output\output>.\"Bubble.exe"

=====
ORDENACAO DE TEMPOS DE ONIBUS (BUBBLE SORT)
=====
Por favor, insira o tempo de chegada (em minutos) de cada onibus:

Digite o tempo do onibus 1: 1

```

WelcomeBubble.c

```
C Bubble.c > main()
7 int main() {
8     // Ordenacao de tempos em ordem crescente
82     for (i = 0; i < QTD_ONIBUS; i++) {
83         printf("%d min | ", tempos[i]);
84     }
85     printf("\n\n");
86
87     return 0;
88 }
```

PROBLEMSOUTPUTTERMINALPORTSDEBUG CONSOLE

=====

ORDENACAO DE TEMPOS DE ONIBUS (BUBBLE SORT)

=====

Por favor, insira o tempo de chegada (em minutos) de cada onibus:

Digite o tempo do onibus 1: 50

Digite o tempo do onibus 2: 42

Digite o tempo do onibus 3: 37

Digite o tempo do onibus 4: 25

Digite o tempo do onibus 5: 58

Digite o tempo do onibus 6: 73

Digite o tempo do onibus 7: 51

Digite o tempo do onibus 8: 28

Digite o tempo do onibus 9:

C/C++ Compile Ru... ✓

C/C++ Compile Run

Add Context...Bubble.c

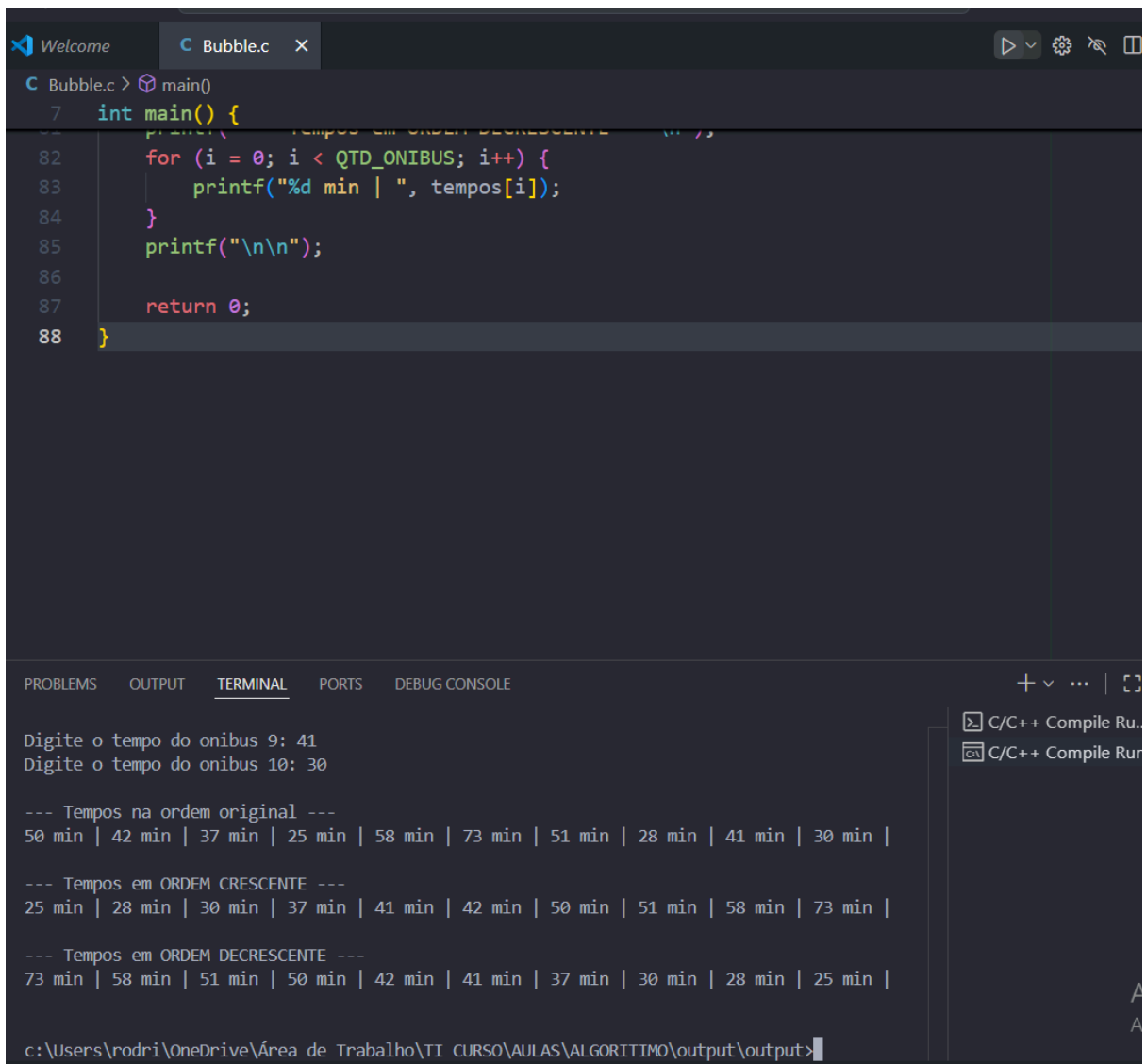
Add context (#), extensions (@), command (C) or file (F) to context menu.

Ativar o Windows

Ative o Windows para obter o melhor desempenho e recursos. Acesse Configurações para ativar o Windows.

Ln 88, Col 2Spaces: 4UTF-8LFGo LiveWin32Prettier

10:3130/09/2025



```

Bubble.c > main()
7  int main() {
81  printf("\n--- Tempos em ORDEM DECRESCENTE ---\n");
82  for (i = 0; i < QTD_ONIBUS; i++) {
83      printf("%d min | ", tempos[i]);
84  }
85  printf("\n\n");
86
87  return 0;
88 }
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

Digite o tempo do onibus 9: 41
 Digite o tempo do onibus 10: 30

--- Tempos na ordem original ---
 50 min | 42 min | 37 min | 25 min | 58 min | 73 min | 51 min | 28 min | 41 min | 30 min |

--- Tempos em ORDEM CRESCENTE ---
 25 min | 28 min | 30 min | 37 min | 41 min | 42 min | 50 min | 51 min | 58 min | 73 min |

--- Tempos em ORDEM DECRESCENTE ---
 73 min | 58 min | 51 min | 50 min | 42 min | 41 min | 37 min | 30 min | 28 min | 25 min |

c:\Users\rodri\OneDrive\Área de Trabalho\TI CURSO\AULAS\ALGORITIMO\output\output>