

Universidade Cruzeiro do Sul | Análise e Desenvolvimento de Sistemas

Aluno: Leandro da Rocha Ferreira

RGM: 43435157

Matéria: Algoritmos e Pensamento Computacional

Experiência prática IV

Código-fonte do programa em C

- **IDE:** Vs Code

```
/*  
  
 * Descrição: Este programa lê 10 tempos de chegada de ônibus,  
  
 * exibe a lista original, ordena em ordem crescente e  
  
 * decrescente usando Bubble Sort, e exibe os resultados.  
  
 */  
  
#include <stdio.h> // Funções de entrada e saída  
  
#include <stdlib.h> // EOF (End of File) na validação  
  
// Constante para o tamanho do vetor.  
  
#define TAMANHO_VETOR 10  
  
int main() {  
  
    // === PASSO 1: DECLARAÇÃO DAS VARIÁVEIS ===  
  
    // Vetor para armazenar os tempos (em minutos) de cada ônibus.  
  
    int tempos[TAMANHO_VETOR];  
  
    // Variável temporária que será essencial para a operação de  
TROCA (swap) durante a ordenação.  
  
    int temp;
```

```
// === PASSO 2: ENTRADA DE DADOS ===

printf("--- Entrada de Dados: Tempo de Chegada dos Onibus (em minutos) ---\n");

printf("Por favor, insira os %d tempos de deslocamento.\n\n", TAMANHO_VETOR);

// Laço 'for' para percorrer cada posição do vetor e pedir um valor.

for (int i = 0; i < TAMANHO_VETOR; i++) {

    // Loop de validação para garantir que o usuário digite um número válido.

    while (1) {

        printf("Digite o tempo do onibus %d: ", i + 1);

        // Tenta ler um número inteiro. Se for bem-sucedido, scanf retorna 1.

        if (scanf("%d", &tempos[i]) == 1) {

            break; // Se a leitura foi válida, sai do loop 'while(1)'.

        }

    }

}
```

```
        // Se o scanf falhou, o programa chega aqui.

        printf("[ERRO] Por favor, digite apenas numeros
inteiros.\n");

        // Limpa o buffer de entrada para evitar um loop
infinito de erro.

        // Lê e descarta todos os caracteres até encontrar um
'\n' (Enter).

        int c;

        while ((c = getchar()) != '\n' && c != EOF);

    }

}

// === PASSO 3: EXIBIÇÃO DO VETOR ORIGINAL ===

printf("\n\n--- Vetor Original (Ordem de Insercao) ---\n");

for (int i = 0; i < TAMANHO_VETOR; i++) {

    printf("%d ", tempos[i]);

}

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

```
// === PASSO 4: ORDENAÇÃO EM ORDEM CRESCENTE (BUBBLE SORT) ===

printf("... Processando ordenacao crescente ...\n");

// Controla o número de "passagens" completas pelo vetor.

for (int i = 0; i < TAMANHO_VETOR - 1; i++) {

    // Percorre o vetor comparando os pares adjacentes.

    for (int j = 0; j < TAMANHO_VETOR - i - 1; j++) {

        // Condição de troca para ORDEM CRESCENTE:

        // Se o elemento da esquerda for MAIOR que o da
direita...

        if (tempos[j] > tempos[j + 1]) {

            // ...eles precisam ser trocados de lugar.

            // 1. Guarda o valor da esquerda em 'temp'.

            temp = tempos[j];

            // 2. Copia o valor da direita para a posição da
esquerda.
```

```

        tempos[j] = tempos[j + 1];

        // 3. Pega o valor original da esquerda (guardado
em 'temp') e coloca na direita.

        tempos[j + 1] = temp;

    }

}

}

// === PASSO 5: EXIBIÇÃO DO VETOR EM ORDEM CRESCENTE ===

printf("\n--- Vetor Ordenado (Crescente) ---\n");

for (int i = 0; i < TAMANHO_VETOR; i++) {

    printf("%d ", tempos[i]);

}

printf("\n\n");

// === PASSO 6: ORDENAÇÃO EM ORDEM DECRESCENTE (BUBBLE SORT)
===

printf("... Processando ordenacao decrescente ...\n");

```

```
for (int i = 0; i < TAMANHO_VETOR - 1; i++) {

    for (int j = 0; j < TAMANHO_VETOR - i - 1; j++) {

        // Se o elemento da esquerda for MENOR que o da
direita...

        if (tempos[j] < tempos[j + 1]) {

            // ...a troca acontece.

            temp = tempos[j];

            tempos[j] = tempos[j + 1];

            tempos[j + 1] = temp;

        }

    }

}

// === PASSO 7: EXIBIÇÃO DO VETOR EM ORDEM DECRESCENTE ===

printf("\n--- Vetor Ordenado (Decrescente) ---\n");
```

```
for (int i = 0; i < TAMANHO_VETOR; i++) {  
  
    printf("%d ", tempos[i]);  
  
}  
  
printf("\n\n");  
  
return 0;  
}
```

Evidências da execução do programa

1. Entrada de dados com erro:

```
--- Entrada de Dados: Tempo de Chegada dos Onibus (em minutos) ---  
Por favor, insira os 10 tempos de deslocamento.  
  
Digite o tempo do onibus 1: 1  
[ERRO] Por favor, digite apenas numeros inteiros.  
Digite o tempo do onibus 1: 1  
[ERRO] Por favor, digite apenas numeros inteiros.  
Digite o tempo do onibus 1: 1  
Digite o tempo do onibus 2: 2  
Digite o tempo do onibus 3: █
```

2. Entrada de dados:


```
--- Entrada de Dados: Tempo de Chegada dos Onibus (em minutos) ---  
Por favor, insira os 10 tempos de deslocamento.
```

```
Digite o tempo do onibus 1: 2  
Digite o tempo do onibus 2: 5  
Digite o tempo do onibus 3: 2  
Digite o tempo do onibus 4: 3  
Digite o tempo do onibus 5: 5  
Digite o tempo do onibus 6: 8  
Digite o tempo do onibus 7: 7  
Digite o tempo do onibus 8: 9  
Digite o tempo do onibus 9: 9  
Digite o tempo do onibus 10: 9
```

```
--- Vetor Original (Ordem de Insercao) ---  
2 5 2 3 5 8 7 9 9 9
```

```
... Processando ordenacao crescente ...
```

```
--- Vetor Ordenado (Crescente) ---  
2 2 3 5 5 7 8 9 9 9
```

```
... Processando ordenacao decrescente ...
```

```
--- Vetor Ordenado (Decrescente) ---  
9 9 9 8 7 5 5 3 2 2
```