# **CURSO DE ENGENHARIA DE SOFTWARE**

# Disciplina: Professor(a): Wagner Oliveira de Araujo, Me Nome do(a) Acadêmico(a): Nome do(a) Acadêmico Completo Disciplina: Ferramentas Computacionais Nota: Via AVA Visto:

### Instruções:

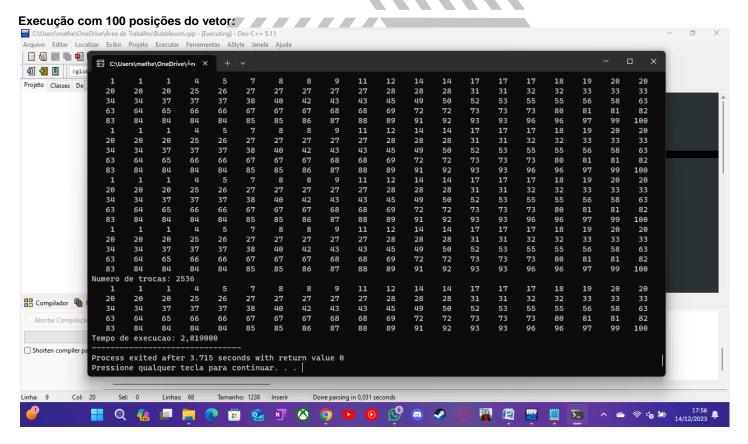
- Coloque seu nome completo no campo Nome do(a) Acadêmico.
- Coloque a data de entrega do trabalho.

Coloque o código aqui juntamente com a impressão de saída do código !!!!!

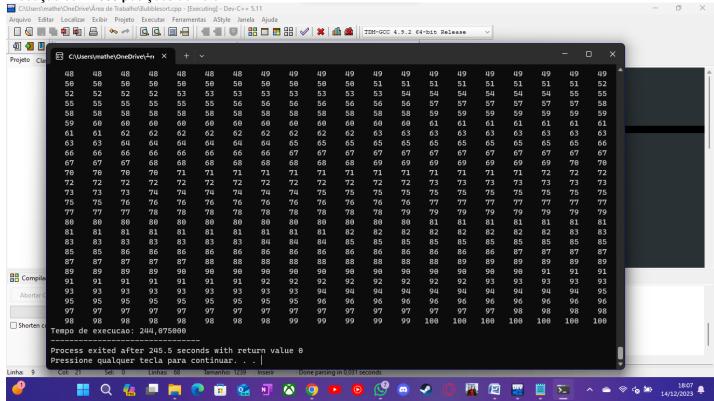
Obs: O código utilizado será um outro modelo de BubbleSort autoral, porém ambos os códigos estarão em um link do GitHub.

```
1) O código e suas observações!!!!
          #include <stdio.h>
          #include <stdlib.h>
          #include <locale.h>
          #include <time.h>
          void gerarVetor(int vet[], int num);
          void imprimirVetor(int vetor[], int num);
          void bubbleSort(int vet[], int num);
          #define tamanho 10
          int main() {
                     setlocale(LC_ALL, "Portuguese");
                     int vetor[tamanho];
                     gerarVetor(vetor, tamanho);
                     clock_t t;
                     bubbleSort(vetor, tamanho);
                     imprimirVetor(vetor, tamanho);
                     t = clock() - t;
                     printf("Tempo de execucao: %lf", ((double)t)/((CLOCKS_PER_SEC)));
                     return 0;
          void gerarVetor(int vet[], int num) {
           srand(time(NULL));
                     for (int i = 0; i < num; i++) {
                                vet[i] = rand() % 100 + 1;
          void imprimirVetor(int vet[], int num) {
                                for (int i = 0; i < num; i++) {
                                           printf("%5d ", vet[i]);
                                printf("\n");
          }
```

O código está em C++, possui uma função para gerar vetor, uma para fazer a ordenação do algoritmo BubbleSort e outra para imprimir o vetor, o tamanho do vetor é definido pela constante, #define tamanho 10 e para cada teste será modificada de 10 para 100 e depois de 100 para 1000. E a cada teste é feita uma nova compilação, o vetor é gerado com valores aleatórios.



### Execução com 1000 posições do vetor:



## 2) Quadro contendo o número de posições

| Vetor e número de posições | Número de Instruções  | Tempo de execução    |
|----------------------------|---|----------------------|
| Vetor[10]                  | O código em sim possui 17 instruções.   | 0,028000 segundos    |
| Vetor[100]                 | Ao alterar o numero do tamanho do vetor para 100 o código mantém as 17 instrucões.  | 2,819000 segundos.   |
| Vetor[1000]                | Ao alterar o numero do tamanho do vetor para 1000 o código mantém as 17 instruções, porém demorou muito mais tempo de execução. Ao aumentar o tamanho do vetor, para ordenar os dados a quantidade de trocas pelo for do vetor aumentou proporcionalmente, não sendo possível contar manualmente. | 244,075000 segundos. |

Link do GitHub com trabalho e os códigos: https://github.com/WarWolfOne/BubbleSort