



**Universidade Federal da Bahia - UFBA**  
**Instituto de Computação - IC**  
**Departamento de Ciência da Computação - DCC**  
**MATA40 – Estrutura de Dados**



**Prof. Ricardo Rios**

**Data:**

**Aluno:**

**3ª Prova**

**Instruções**

- Leia atentamente todas as questões;
- Todas as respostas devem ser implementadas e submetidas via run.codes;
- Alunos envolvidos em plágio terão notas zeradas (tanto quem forneceu o código quanto quem copiou) - A ferramenta run.codes detecta plágios!
- As entradas apresentadas em cada questão são ilustrativas, servem apenas para exemplificar o formato de entrada dos dados, e devem variar dependendo dos casos de teste no run.codes.

**1** Implemente um algoritmo que recebe a seguinte entrada no run.codes:

7  
3  
10  
5  
9  
8  
1  
3

A primeira linha informa quantas linhas você deve ler na sequência para executar o seu código. A segunda linha informa o número máximo de iterações que seu código deve executar. As demais linhas são os dados que precisam ser ordenados. Em seguida, seu algoritmo deve retornar os dados (10 5 9 8 1 3) ordenados usando o algoritmo de Seleção até a iteração 3 (segunda linha). Exemplo:

Dados de entrada: 10 5 9 8 1 3

Iteração 01:

1 5 9 8 10 3

Iteração 02:

1 3 9 8 10 5

Iteração 03:

1 3 5 8 10 9 (retorno da função)

- 2 Repita o exercício anterior, mas implementando o algoritmo de ordenação por inserção. Exemplo:

Dados de entrada: 10 5 9 8 1 3

Iteração 01:

10

Iteração 02:

5 10

Iteração 03:

5 9 10 (retorno da função)

- 3 Repita o exercício anterior, agora implementando o algoritmo de ordenação Bubblesort. Exemplo:

Dados de entrada: 10 5 9 8 1 3

Iteração 01:

5 9 8 1 3 10

Iteração 02:

5 8 1 3 9 10

Iteração 03:

5 1 3 8 9 10 (retorno da função)

- 4 Implemente o algoritmo de ordenação Quicksort, considerando a seguinte entrada:

8

1

10

5

9

8

1

3

2

A primeira linha informa quantas linhas você deve ler na sequência para executar o seu código. A segunda linha informa qual é a posição (começando da posição 0) do elemento que deve ser usado como pivô na primeira iteração. Nas iterações seguintes, o primeiro valor deve ser escolhido. As demais linhas são os dados que precisam ser ordenados. Seu algoritmo deve executar todas as etapas do algoritmo e imprimir o resultado da ordenação. Exemplo:

Dados de entrada: 10 5 9 8 1 3 2

Iteração 01:

[ 1 3 2 5 10 9 8 ]

Iteração 02:

[ 1 3 2 ] 5 [ 10 9 8 ]

(...)

Resultado final: 1 2 3 5 8 9 10

**5** Implemente um algoritmo que lê a seguinte entrada:

```
7
4
3
computacao
estrutura
ondina
ufba
dados
```

A primeira linha informa quantas linhas você deve ler na sequência para executar o seu código. A segunda linha informa qual é o tamanho da tabela hash (número de posições). O número na terceira linha serve para dizer qual posição da hash deve ser impressa (posição começando do número 0). Na sequência, há palavras que devem ser armazenadas numa hash. A função para codificação da palavra deve converter cada letra em um número de acordo com sua posição no alfabeto e somá-las. Exemplo:

$ufba \rightarrow 'u', 'f', 'b', 'a' \rightarrow 21, 6, 2, 1 \rightarrow 30$

O resultado do somatório será a posição na hash usando o algoritmo de multiplicação visto em sala de aula:

Posição: 4

No exemplo, a hash ficaria:

```
[0] - computacao
[1] - ondina
[2] - ufba
[3] - estrutura -> dados
```

Por fim, a saída do código deve ser:

estrutura dados