



Pontifícia Universidade Católica de Minas Gerais  
Curso de Ciência da Computação - Coração Eucarístico  
Avaliação de Desempenho Acadêmico (ADA)

# Prova II - 13:30 às 16:20

Aluno: \_\_\_\_\_

## 1 DIAGONAL PRINCIPAL DA MATRIZ

Em álgebra linear, a diagonal principal de uma matriz  $A$  é a coleção das entradas  $A_{i,j}$  em que  $i$  é igual a  $j$ . A diagonal principal de uma matriz quadrada une o seu canto superior esquerdo ao canto inferior direito.

**Dados de entrada:** Como entrada do programa, o texto contém várias linhas sendo que a primeira contém um número inteiro  $N$  indicando o número de linhas/colunas de nossa matriz quadrada. Em seguida, as próximas  $N$  linhas possuem  $N$  colunas de inteiros com os elementos da matriz.

**Exemplo de entrada:**

```
3
1 2 3
4 5 6
7 8 9
```

**Dados de saída:** A saída produzida pelo programa é simples. Ela contém uma linha com  $N$  colunas indicando os elementos da diagonal principal da matriz de entrada.

**Exemplo de saída:**

```
1 5 9
```

## 2 ENCONTRAR O MAIOR VALOR

Um problema importante na Ciência da Computação consiste em dado um conjunto de valores, encontrar o maior deles.

**Dados de entrada:** Como entrada do programa, o texto contém várias linhas e, em cada uma delas, um conjunto de números sendo que o primeiro valor corresponde a quantidade de números na linha. A última linha contém apenas o número 0.

**Exemplo de entrada:**

```
8 1 2 3 4 5 6 7 0
9 9 8 7 6 5 4 3 2 1
6 5 3 2 4 6 7
9 9 7 6 4 2 1 8 5 3
3 1 2 3
3 0 2 1
0
```

**Dados de saída:** A saída produzida pelo programa é simples. Para cada conjunto de entrada, o maior valor é exibido.

**Exemplo de saída:**

```
7
9
7
9
3
2
```

### 3 LISTAS

As listas são um dos principais tipos de estruturas de dados onde podemos inserir elementos em qualquer posição da mesma. Podemos considerar que nossa lista tem sete operações básicas:

- **Inserir no início (II)** que insere um elemento na primeira posição de nossa estrutura.
- **Remover do início** que remove o primeiro elemento de nossa estrutura.
- **Inserir no fim (IF)** que insere um elemento no final de nossa estrutura (na verdade na primeira posição livre).
- **Remover do fim (IF)** que remove o último elemento de nossa estrutura.
- **Inserir qualquer (I)** que insere um elemento em uma posição qualquer (válida) recebida como parâmetro.
- **Remover qualquer (R)** que remove o elemento existente em uma posição qualquer recebida como parâmetro.
- **Mostrar (M)** que mostra os elementos existentes em nossa estrutura.

**Dados de entrada:** Como entrada do programa, o texto contém várias linhas e, em cada uma delas, um comando a ser executado em nossa lista. A última linha contém apenas o número 0. Destaca-se que os comandos de inserção possuem um número inteiro indicando o valor a ser inserido na lista. O comando Inserir qualquer (I) tem dois inteiros, respectivamente, o valor a ser inserido e a posição de inserção. O comando de Remover qualquer (R) tem um inteiro indicando a posição de remoção. Finalmente, o comando de Mostrar (M) apenas exibe os elementos da lista na tela.

**Exemplo de entrada:**

```
II 1
IF 2
I 3 1
M
IF 4
IF 5
RI
RF
R 1
M
0
```

**Dados de saída:** A saída produzida pelo programa é simples. Corresponde as execuções do comando de mostrar.

**Exemplo de saída:**

```
1 3 2
3 4
```

## 4 ESCADINHA

Dizemos que uma sequência de números é uma escadinha, se a diferença entre números consecutivos é sempre a mesma. Por exemplo, “2, 3, 4, 5” e “10, 7, 4” são escadinhas. Note que qualquer sequência com apenas um ou dois números também é uma escadinha!

Neste problema estamos procurando escadinhas em uma sequência maior de números. Dada uma sequência de números, queremos determinar quantas escadinhas existem. Mas só estamos interessados em escadinhas tão longas quanto possível. Por isso, se uma escadinha é um pedaço de outra, consideramos somente a maior. Por exemplo, na sequência “1, 1, 1, 3, 5, 4, 8, 12” temos 4 escadinhas diferentes: “1, 1, 1”, “1, 3, 5”, “5, 4” e “4, 8, 12”.

**Dados de entrada:** A entrada é composta por vários casos de teste. Uma linha da entrada contém um inteiro  $N$  indicando o tamanho da sequência de números. A linha seguinte contém  $N$  inteiros definindo a sequência. A última linha contém a frase “FIM”.

**Exemplo de entrada:**

```
8
1 1 1 3 5 4 8 12
1
112
5
11 -106 -223 -340 -457
FIM
```

**Dados de saída:** A saída produzida pelo programa é simples. Imprima uma linha contendo um inteiro representando quantas escadinhas existem em cada sequência.

**Exemplo de saída:**

```
4
1
1
```