

## Matrizes especiais

Você pode utilizar qualquer ambiente de programação para desenvolver sua atividade. Ao final, **copie e cole o seu código-fonte com a resposta aqui mesmo neste documento**, dentro dos espaços indicados para isso e **preservando a indentação do código**. **Depois que terminar sua avaliação, não se esqueça de entregar sua atividade!** Fique atento ao relógio, pois as atividades entregues com atraso não serão aceitas.

A matriz ampulheta com sinal invertido é uma matriz quadrada de ordem  $n$  em que há elementos não nulos na primeira e última linhas, na diagonal principal e na diagonal secundária. Além disso, os elementos da diagonal principal são os mesmos da diagonal secundária, linha-a-linha, mas com o sinal trocado, e os elementos da primeira linha são os mesmos da última linha, mas com o sinal trocado. A exceção é o elemento do cruzamento entre as diagonais principal e secundária, caso a ordem da matriz seja ímpar. Os demais elementos da matriz são iguais a zero. Assume-se aqui que esse tipo de matriz possui ao menos ordem 3. Veja a seguir um exemplo deste tipo de matriz,  $A_{5 \times 5}$ :

$$A_{5 \times 5} = \begin{bmatrix} 4 & 2 & 5 & -8 & -4 \\ 0 & -7 & 0 & 7 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ -4 & -2 & -5 & 8 & 4 \end{bmatrix}$$

O TAD `MatrizEspecial` representa esse tipo de matriz. Os valores devem ser armazenados no TAD `MatrizEspecial` por meio de uma representação linear com um único vetor (`vet`) e de modo que a quantidade de elementos armazenados seja mínima. Para esse TAD, desenvolver:

- A) Construtor (que recebe a ordem  $n$  da matriz como parâmetro) e destrutor da classe. Não esquecer de indicar no construtor a **forma** de representação dos elementos não nulos da matriz. Imprimir a mensagem de erro “Dimensões inválidas” e sair do programa, caso o valor de  $n$  não seja válido para esse tipo de matriz.

```
// Cole aqui sua resposta
```

- B) A operação `int detInd(int i, int j)` para verificar se os índices  $i$  e  $j$  da matriz são válidos (retornar -1 caso não sejam), e determinar o índice do vetor em que se encontra o elemento na posição indicada por  $i$  e  $j$ . Retornar -2 se os índices representam uma posição de valor zero. Use outras *flags* se necessário.

```
// Cole aqui sua resposta
```

- C) A operação pública `float get(int i, int j)` para retornar o valor da posição `i` e `j` da matriz. Imprimir a mensagem de erro “Índices inválidos” e sair do programa, caso os índices não representem uma posição válida para a matriz.

```
// Cole aqui sua resposta
```

- D) A operação pública `void set(int i, int j, float val)` para atribuir o valor na posição `i` e `j` da matriz. Emitir a mensagem de erro: “Tentando atribuir valor nao zero em posição impropria”, caso o usuário tente atribuir um valor diferente de zero na posição que deve ser zero. Imprimir a mensagem de erro “Índices inválidos”, caso os índices não representem uma posição válida para a matriz. Note que aqui não há a instrução de saída do programa no caso de erro.

```
// Cole aqui sua resposta
```