

Problema C

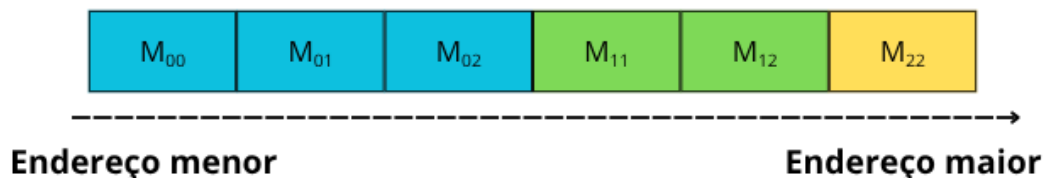
Uma matriz triangular superior só existe da diagonal principal para cima. Essa matriz pode ser representada por **armazenamento por linha**, os elementos da matriz são organizados sequencialmente, linha por linha, em um único vetor contíguo na memória.

Imagine uma matriz triangular superior 3x3 (3 linhas e 3 colunas).

M_{00}	M_{01}	M_{02}
	M_{11}	M_{12}
		M_{22}

No armazenamento por linha, guardamos todos esses elementos em um único vetor, colocando primeiro todos os elementos da linha 0, depois todos da linha 1, e assim por diante:

Memória RAM



Para acessar um elemento na posição $[i][j]$ (linha i , coluna j) em uma matriz com `num_colunas` colunas, precisamos descobrir sua posição no vetor unidimensional.

A ideia é:

- Cada linha completa ocupa um número variável de posições no vetor (depende da linha – linha 1 tem 1 elemento, linha 2 tem 2 elementos ...)
- Para chegar até a linha i , precisamos "pular" todas as linhas anteriores
- Depois, avançamos j posições dentro dessa linha

Portanto, a posição no vetor será uma combinação que leva em conta:

1. Quantas linhas completas existem antes da linha desejada
2. Quantas posições devemos avançar dentro da linha atual

Sua tarefa é implementar a seguinte função no arquivo *matriz.c*:

- `void atualizaElemento(Matriz *m, int linha, int coluna, int valor)`: Como se fizesse a operação `m[linha][coluna] = valor`
- `int recuperaElemento(Matriz *m, int linha, int coluna)`: Como se fizesse a operação `valor = m[linha][coluna]`

```
cd ~  
git clone https://github.com/elvertton/LabProgII/  
cd LabProgII/2025-2/Exercicios-8-Filas-Matrizes/T2/ProblemaC
```

No sistema BOCA, você deverá submeter apenas o arquivo *matriz.c*.

Entrada

A primeira linha contém um único número inteiro representando a dimensão da matriz quadrada. As linhas seguintes contém os elementos da matriz triangular superior, organizados linha por linha. Cada linha da entrada corresponde a uma linha da matriz, com os elementos separados por espaços.

Saída

A matriz triangular superior lida com a parte inferior zerada.

Exemplo de entrada	Exemplo de saída
3 4 5 6 2 3 1	4 5 6 0 2 3 0 0 1