

Considere a seguinte definição de uma matriz numérica e complete as seguintes operações fundamentais:

```
type Row = [Float]
data Matrix = Matrix { ncols :: Int
                      , rows :: Int
                      , rows :: [Row]
                      }
```

matriz de zeros

```
zeroMatrix :: Int → Int → Matrix
```

matriz de uns

```
oneMatrix :: Int → Int → Matrix
```

matriz identidade : recebe ordem

```
identMatrix :: Int → Matrix
```

soma duas matrizes

```
sumMatrix :: Matrix → Matrix → Matrix
```

produto de escalar por matriz

```
prodScalar :: Float → Matrix → Matrix
```

produto entre matrizes

```
prodMatrix :: Matrix → Matrix → Matrix
```

transforma listas de listas de floats numa matriz

```
listToMatrix :: [Row] → Matrix
```

instance Show Matrix where

show = undefined

A sua definição deve produzir os seguintes resultados:

```
zeroMatrix 2 2
```

```
| 0.0 0.0 |
| 0.0 0.0 |
```

```
oneMatrix 2 3
```

```
| 1.0 1.0 1.0 |
| 1.0 1.0 1.0 |
```

```
identMatrix 3
```

```
| 1.0 0.0 0.0 |
| 0.0 1.0 0.0 |
| 0.0 0.0 1.0 |
```

```
m1
```

```
| 1.0 2.0 |
| 3.0 4.0 |
```

```
m2
```

```
| 3.0 4.0 |
| 6.0 7.0 |
```

```
sumMatrix m1 m2
```

```
| 4.0 6.0 |
| 9.0 11.0 |
```

```

prodScalar 2 m1
| 2.0 4.0 |
| 6.0 8.0 |
prodMatrix m1 m2
| 15.0 18.0 |
| 33.0 40.0 |
prodMatrix m1 m3
| 15.0 18.0 14.0 |
| 33.0 40.0 34.0 |
listToMatrix [[1,2],[3,4]]
| 1.0 2.0 |
| 3.0 4.0 |

```