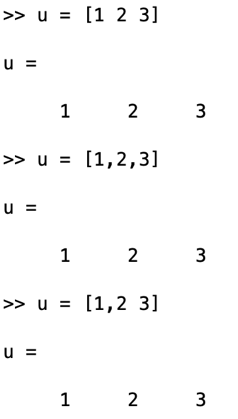
**Notas sobre MatLab – ficha 1 (Erros)**

**Sobre variáveis:**

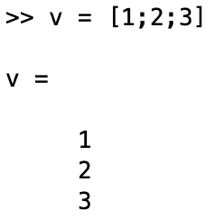
* O primeiro carater de uma variável nunca pode ser um número, não pode ter espaços, carateres usados por operações (+ - / ^)
* Se quisermos separar podemos usar o \_
* Não se podem usar carateres específicos do teclado português (ç ou assentos)
* O MatLab é case sensitive (letra maiúscula é diferente da mesma letra minúscula)
* Não se podem usar variáveis reservadas ao matlab (exp log)
* Nos nºs não podemos usar virgulas mas sim pontos

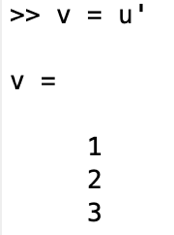
**Vetores ou matrizes:**

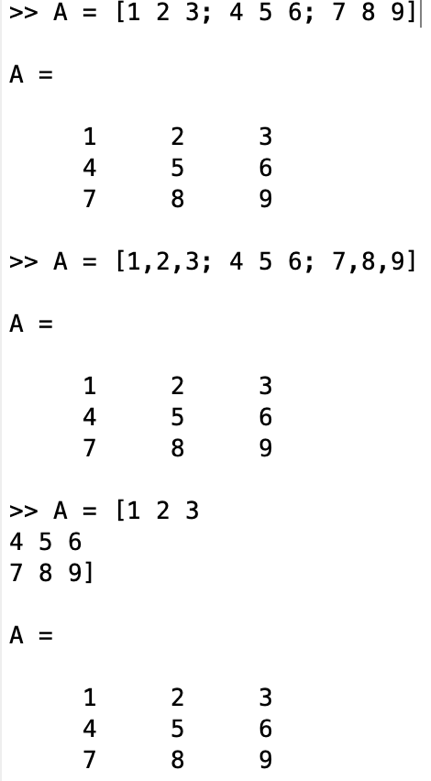
* Deve-se escrever sempre entre parênteses retos
* Separamos os elementos do vetor **linha** por espaços ou por vírgulas (podemos até misturar virgulas e espaços)

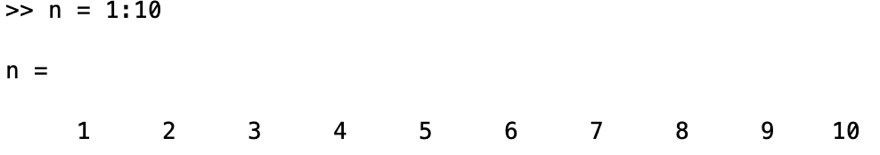


* Separamos os elementos do vetor **coluna** por ponto e virgula



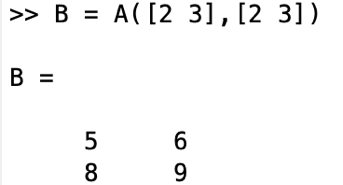
* O vetor transposto de uma linha é uma coluna quando temos os mesmos elementos. No caso o vetor v é o vetor u transposto. **Transposto** de um vetor linha é v = u’
* Para construirmos uma **matriz** usamos a junção de vetores linha com vetores coluna ou então, não fechamos a condição e escrevemos a matriz linha a linha como no último caso.



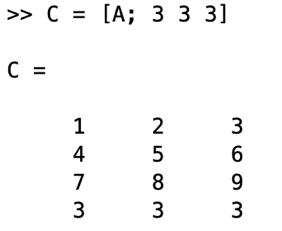
* Vetor linha com os números naturais menores ou iguais a 10. n = 1:10 -> vai originar um **vetor linha que começa em 1, e que vai de um em um até 10**.
* Vetor linha com os números pares naturais menores ou iguais a 12. p = 2:2:12-> vai originar um **vetor linha que começa em 2 (primeiro algarismo), e que vai de 2 em 2 (algarismo do meio) até 12 (último algarismo)**. Estes algarismos não precisam de ser inteiros e quando não o são a virgula é um ponto.

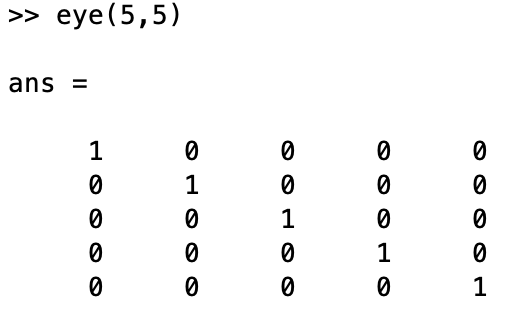
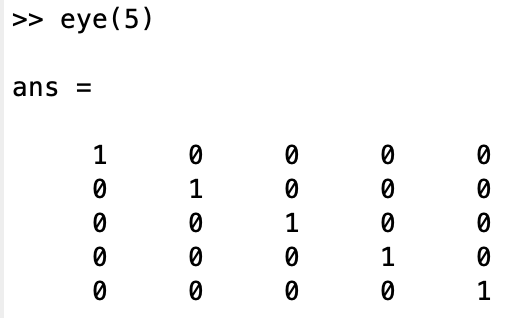
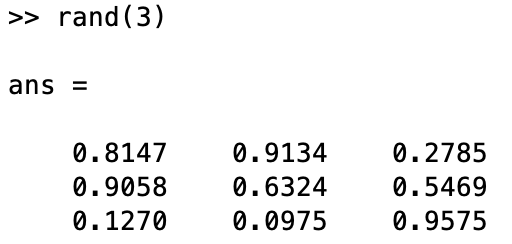


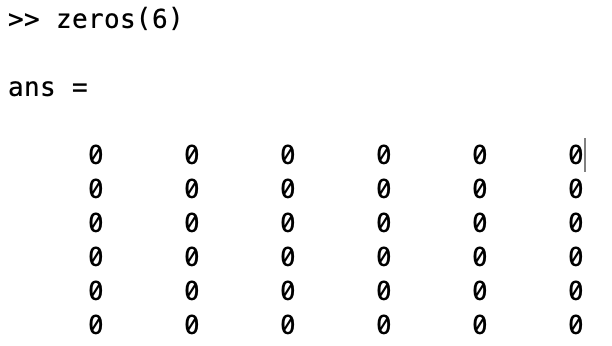
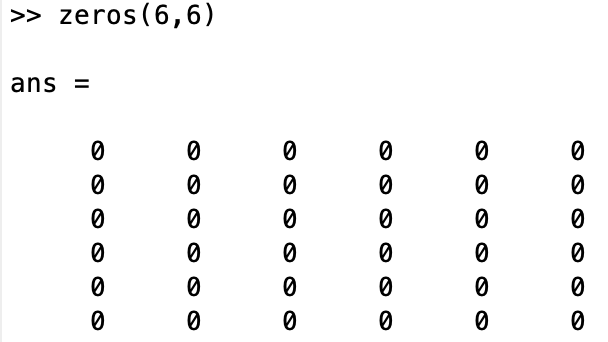
* Uma matriz com os elementos das linhas 2 e 3 (primeiras a escrever) e colunas 2 e 3 (segundas a escrever) da matriz A



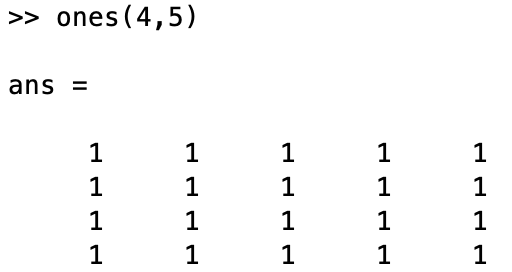
* C = [A; 3 3 3] -> vai **acrescentar** à matriz A **uma linha** (3 3 3).

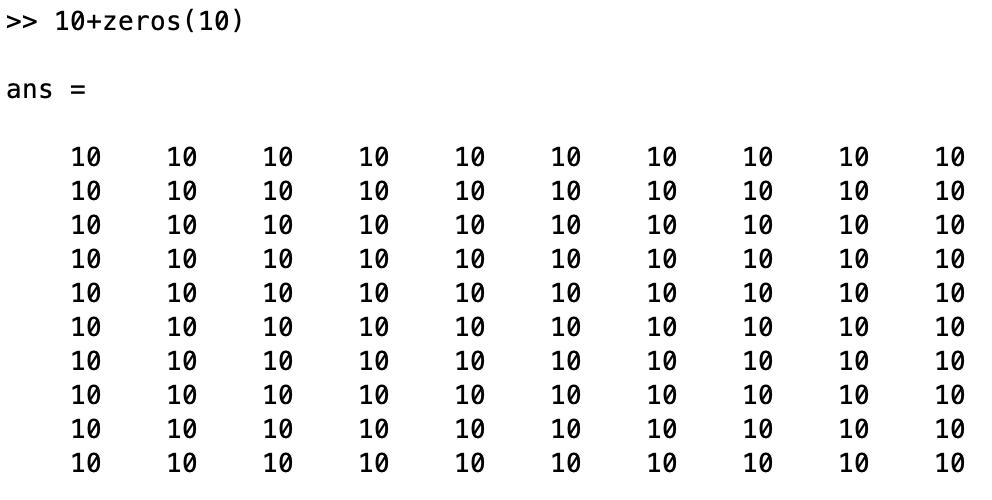
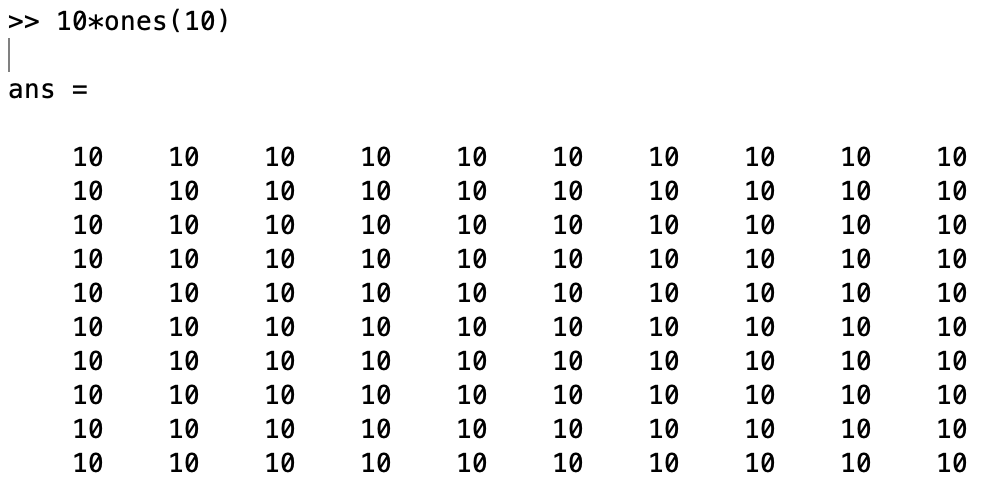
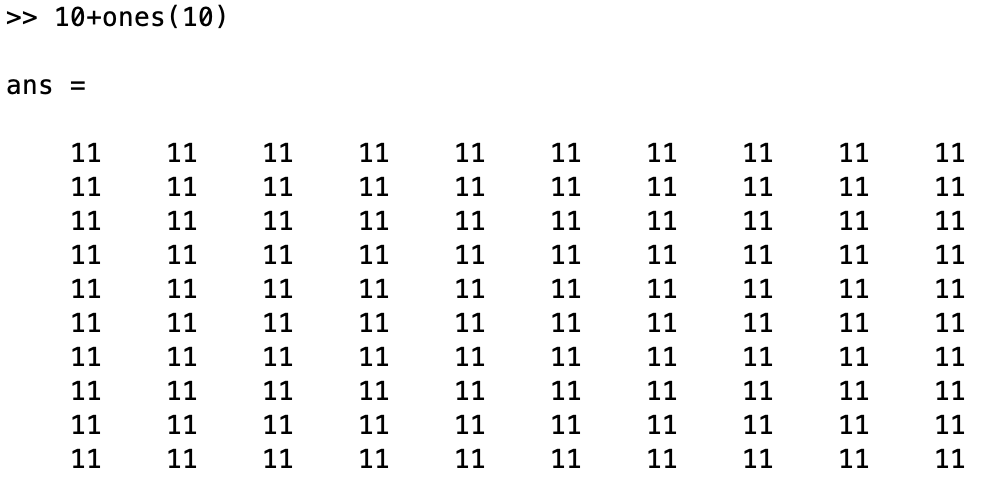


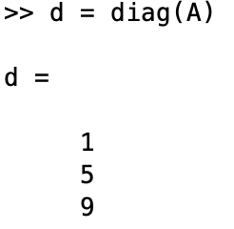
* Como adicionar uma coluna a uma matriz (?)
* Matriz **identidade** 5 linhas (primeiro nº) x 5 colunas (segundo nº). -> eye. Quando é quadrada apenas basta colocarmos uma vez.
* Matriz 3x3 com **números aleatórios entre o 0 e 1** -> rand.
* Matriz 4x3 com números aleatórios entre -1 e 1 (?)
* Matriz **nula** 6 linhas (primeiro nº) x 6 colunas (segundo nº) -> zeros. Quando é quadrada apenas basta colocarmos uma vez.



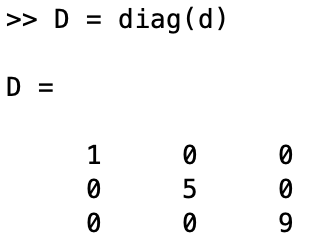
* Matriz 4x5 só com **algarismos 1** -> ones.



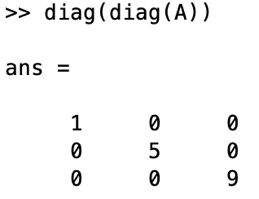
* Matriz 10x10 com elementos 10
* Matriz 10x10 com elementos 11
* **Vetor coluna com os elementos da diagonal** da matriz A -> diag(A)



1. Matriz com os elementos do **vetor d na diagonal e os restantes elementos iguais a zero** -> diag(d)



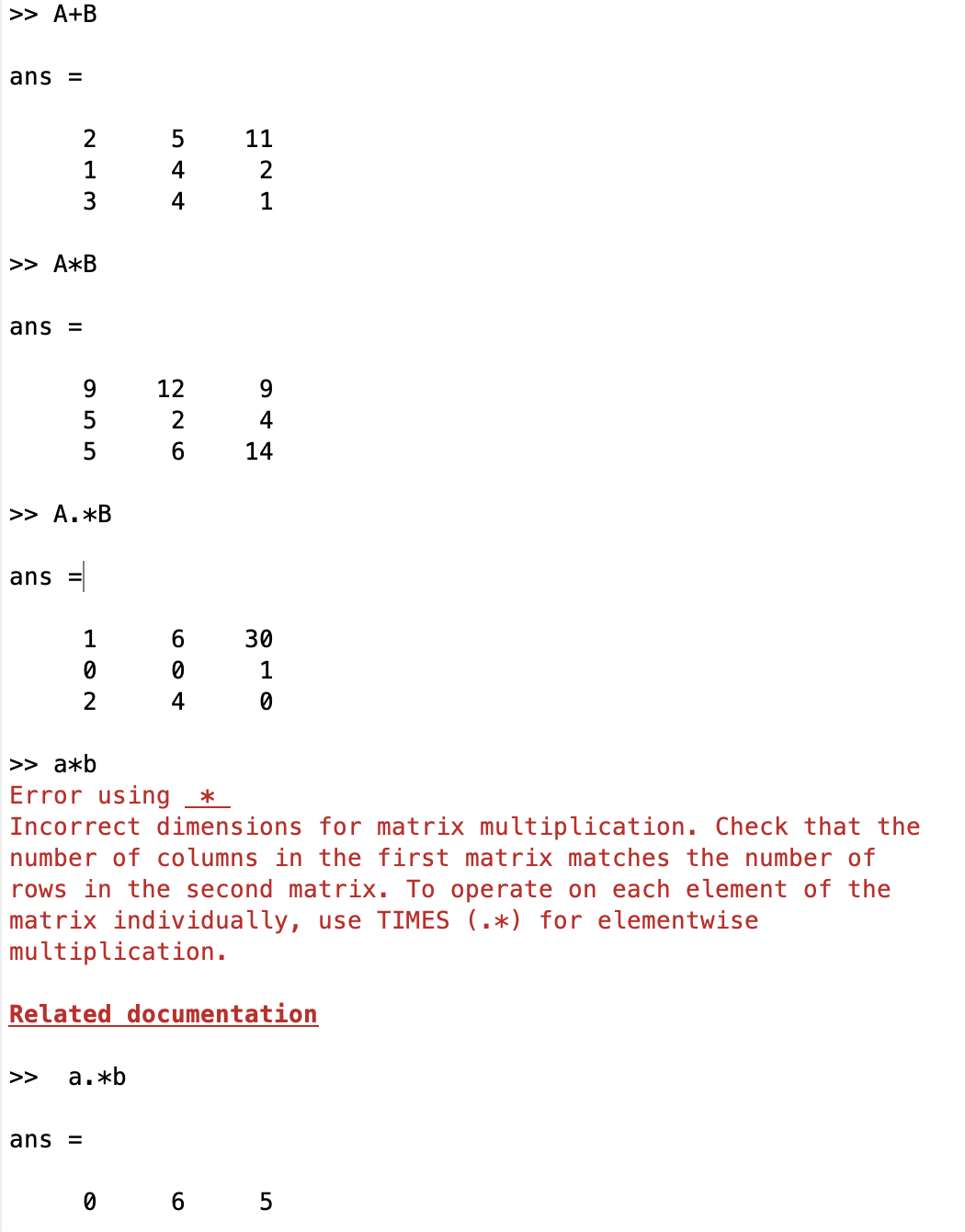
1. Para fazer tudo ao mesmo tempo:



* **Somar todas as colunas de uma matriz** vai dar um vetor linha -> sum(A,1)
* **Somar todas as linhas de uma matriz** vai dar um vetor coluna -> sum(A,2)

**Operações com matrizes** (exercício 3)**:**

\*Não dá para calcular a\*b porque o nº de colunas da primeira matriz não é igual ao nº de linhas da 2º matriz.

a)

b)

c)

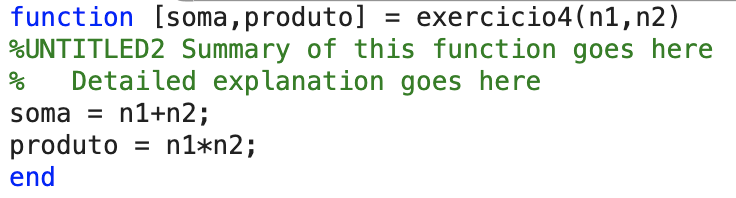
\*e)

d)

Na alínea c) e d) **.\*** representa a multiplicação por cada um dos elementos de a por b

**Funções** (exercício 4)**:**

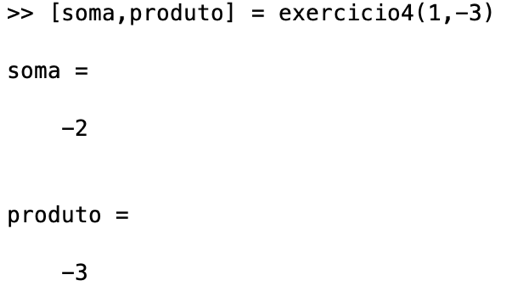
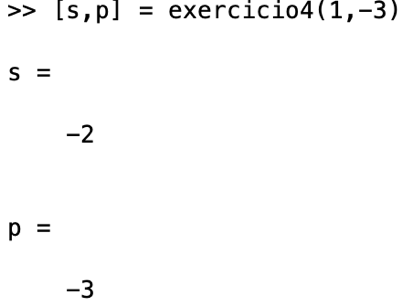
**Para criar uma função:**

New -> function -> dar nome ao untilted -> preencher os parâmetros de saída -> preencher os parâmetros de entrada -> escrever a relação entre os parâmetros de saída e os parâmetros de entrada -> gravar a função no save com o nome que colocamos no untilted

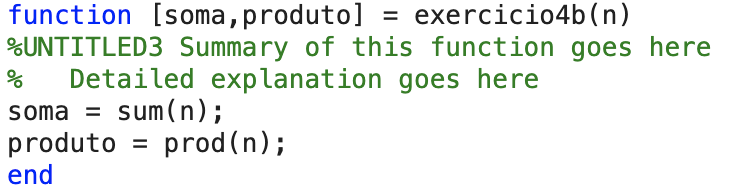
**a)**

**Para usar a função:**

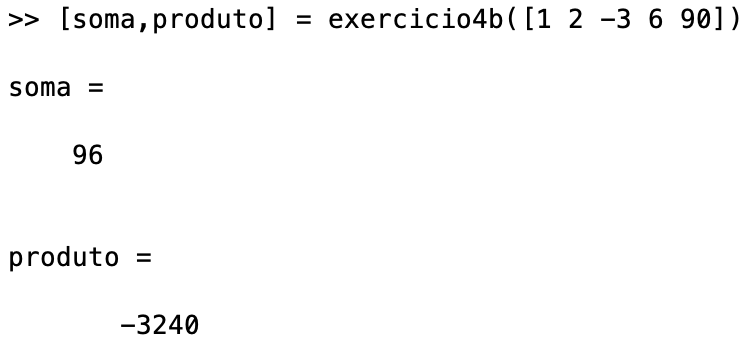
Temos de usar a estrutura com a qual definimos a função e podemos ou não alterar os nomes dos parâmetros de entrada e saída.



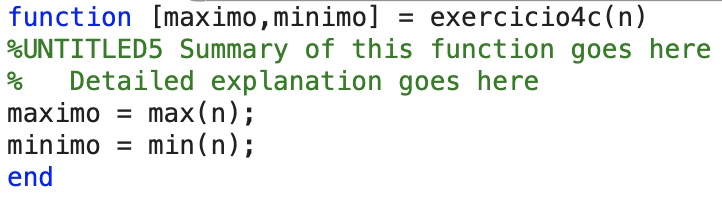
**b)**



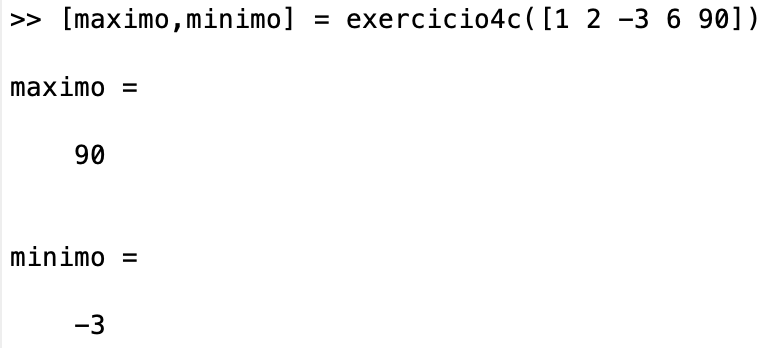
Função:

Chamar a função:

**c)**



Função:



Chamar a função: