

SME0301 - 2-13
Gustavo Carlos Buscaglia

ICMC - Ramal 738176, gustavo.buscaglia@gmail.com

Lista/Prova 1

1. `> a = [1, 2, 3];`
`> b = [4; 2; 1];`
`> a'*b'`

Identifique a resposta que será obtida:

(a) `4 8 12`
`2 4 6`
`1 2 3`

(b) `4`

(c) `11`

(d) `4 2 1`
`8 4 2`
`12 6 3`

(e) **Error. Nonconformant arguments.**

2. No arquivo "a.txt" tem o seguinte

`1.1 2.2 3.3`
`4.4 5.5 6.6`

Você digita:

`> load('a.txt')`
`> b = [100, 200]`

Diga se verdadeiro ou falso:

(a) `> c = [a,b]` não da erro.

(b) `> c = [a',b]` não da erro.

(c) `> size([a;b'])` não da erro.

(d) `> size([a,b'])` não da erro.

(e) `> size([a,b'])`
`ans = 2 4`

3. No arquivo "a.txt" tem o seguinte

`1.1 2.2 3.3`
`4.4 5.5 6.6`

Você digita:

`> load('a.txt')`
`> b = ones(2)`

Diga se verdadeiro ou falso:

(a) `> c = a*b` não da erro.

(b) `> c = a*b - b*a`
`ans = 0`

(c) `> b*a`
`ans =`
`5.5 7.7 9.9`
`5.5 7.7 9.9`

(d) `> max(a)`
`ans = 4.4 5.5 6.6`

(e) `> max(max(a))`
`ans = 2 2 2`

(f) `> [u,v] = max(max(a))`
`u = 6.6`
`v = 3`

4. Você digita

```
> a = [1,2;3,4]
```

```
> b = eye(2)
```

Diga se verdadeiro ou falso:

(a) > c = a*b não da erro.

(b) > c = a*b - b*a

```
c =  
0 0  
0 0
```

(c) > c = a*b - a.*b

```
c =  
0 0  
0 0
```

(d) > max(max(a*b-a.*b))

```
ans = 2
```

(e) > inv(a.*b)

```
ans =  
1.0 0.0  
0.0 0.5
```

(f) > inv(a*b)

```
ans =  
-2.0 1.0  
1.5 -0.5
```

5. Você digita

```
> a = [1,2;3,4]
```

```
> b = eye(2)
```

Diga se verdadeiro ou falso:

(a) > c = a*b não da erro.

(b) > c = a*b - b*a

```
c =  
0 0  
0 0
```

(c) > c = a*b - a.*b

```
c =  
0 0  
0 0
```

(d) > max(max(a*b-a.*b))

```
ans = 2
```

(e) > inv(a.*b)

```
ans =  
1.0 0.0  
0.0 0.5
```

(f) > inv(a*b)

```
ans =  
-2.0 1.0  
1.5 -0.5
```

6. Você digita

```
> x = [0:1:11]
```

```
> v = [1:1:5,3,4,5:-1:1]
```

Diga se verdadeiro ou falso:

- (a) > c = x*v não da erro.
 - (b) > c = x'*v não da erro.
 - (c) > c = x'*v
c = 204.
 - (d) > c = x*v'
c = 204.
 - (e) > c = x'*v;
> c(4,3)
ans = 9.
 - (f) > c = x'*v;
> c(3,4)
ans = 9.
-

7. Com a sequência de comandos:

```
> n = 3; i = n; j = 1;  
> A = (n-1)*ones(n) - eye(n);  
> while true,  
> if i == 0,  
> break;  
> end;  
> if i == j,  
> A(i,j) = A(i,j) - i;  
> else,  
> A(i,j) = i + j;  
> end;  
> i = i - 1;  
> j = n - i;  
> end;  
> A
```

Temos como resultado:

- (a) > A =
0 3 2
3 0 2
4 2 0
 - (b) > A =
1 3 4
3 1 2
2 2 1
 - (c) > A =
1 3 2
3 1 2
4 2 1
 - (d) > A =
4 3 2
3 1 2
3 2 0
 - (e) > error
-

8. Sejam os seguintes comandos:

```
> A = [2 5 6; 7 5 1; 6 8 0];
```

```
> v = [5 1];
```

Diga se verdadeiro ou falso:

- (a) > C = A(:,1)*v não da erro.
- (b) > C = A(3,:)*v não da erro.

- (c) `> sum(A(:,2))`
`ans = 18`
- (d) `> A(length(v), sum(v)-3)`
`ans = 2`
- (e) `> C = [A, [v, 4]']` não da erro.
-

9. Sejam os seguintes comandos:

`> u = 1:1.5:3; v = 1:5;`

Diga se verdadeiro ou falso:

- (a) `> w = [u;v]`; não da erro.
- (b) `> w = [u,v']`; não da erro.
- (c) `> w = [u;v]`
`w =`
1.0 1.0
1.5 2.0
2.0 3.0
2.5 4.0
3.0 5.0
- (d) `> length([u,v]')`
`ans =`
10
- (e) `> size([u';v']')`
`ans =`
10 1
-