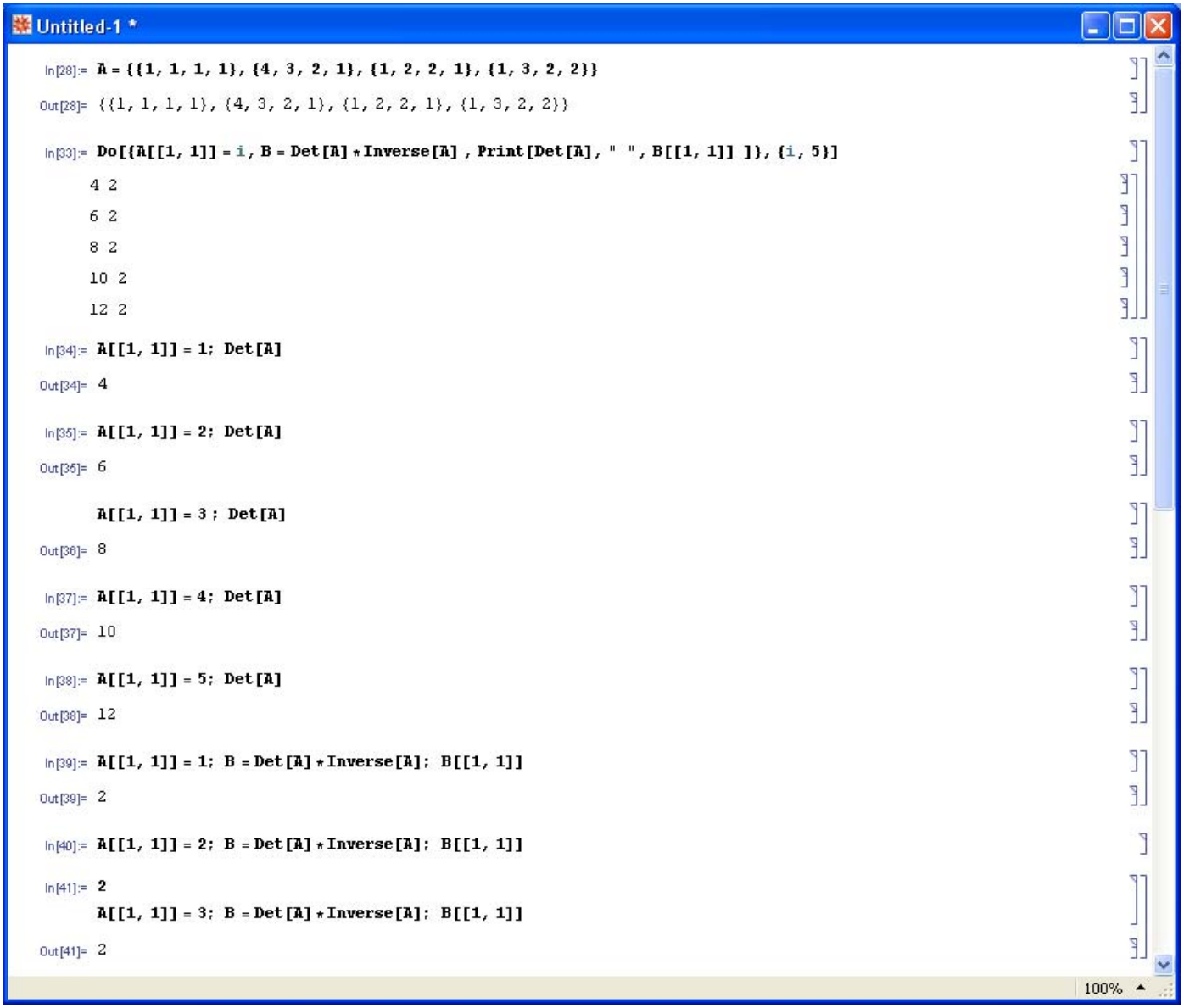
-

Code here is in two parts. Written as a for/do loop and also “calculator” style. The loop method is certainly more efficient coding style. Homework can be done either way.

The scilab code is exactly the same as what one would type in matlab or octave.

```
->A=[1 1 1 1; 4 3 2 1; 1 2 2 1; 1  3 2 2]
 A  =
   1.   1.   1.   1.
   4.   3.   2.   1.
   1.   2.   2.   1.
   1.   3.   2.   2.
-->for j=1:5,     A(1,1)=j;     B=det(A)*inv(A);[det(A)  B(1,1)],    end
 ans  =
   4.   2.
 ans  =
   6.   2.
 ans  =
   8.   2.
 ans  =
   10.   2.
 ans  =
   12.   2.

-->A(1,1)=1;det(A)
 ans  =
   4.
-->A(1,1)=2;det(A)
 ans  =
   6.
-->A(1,1)=3;det(A)
 ans  =
   8.
-->A(1,1)=4;det(A)
 ans  =
   10.
-->A(1,1)=5;det(A)
 ans  =
   12.
-->A(1,1)=1; B=inv(A)*det(A);B(1,1)
 ans  =
   2.
-->A(1,1)=2; B=inv(A)*det(A);B(1,1)
 ans  =
   2.
-->A(1,1)=3; B=inv(A)*det(A);B(1,1)
 ans  =
   2.
-->A(1,1)=4; B=inv(A)*det(A);B(1,1)
 ans  =
   2.
-->A(1,1)=5; B=inv(A)*det(A);B(1,1)
 ans  =
   2.
```

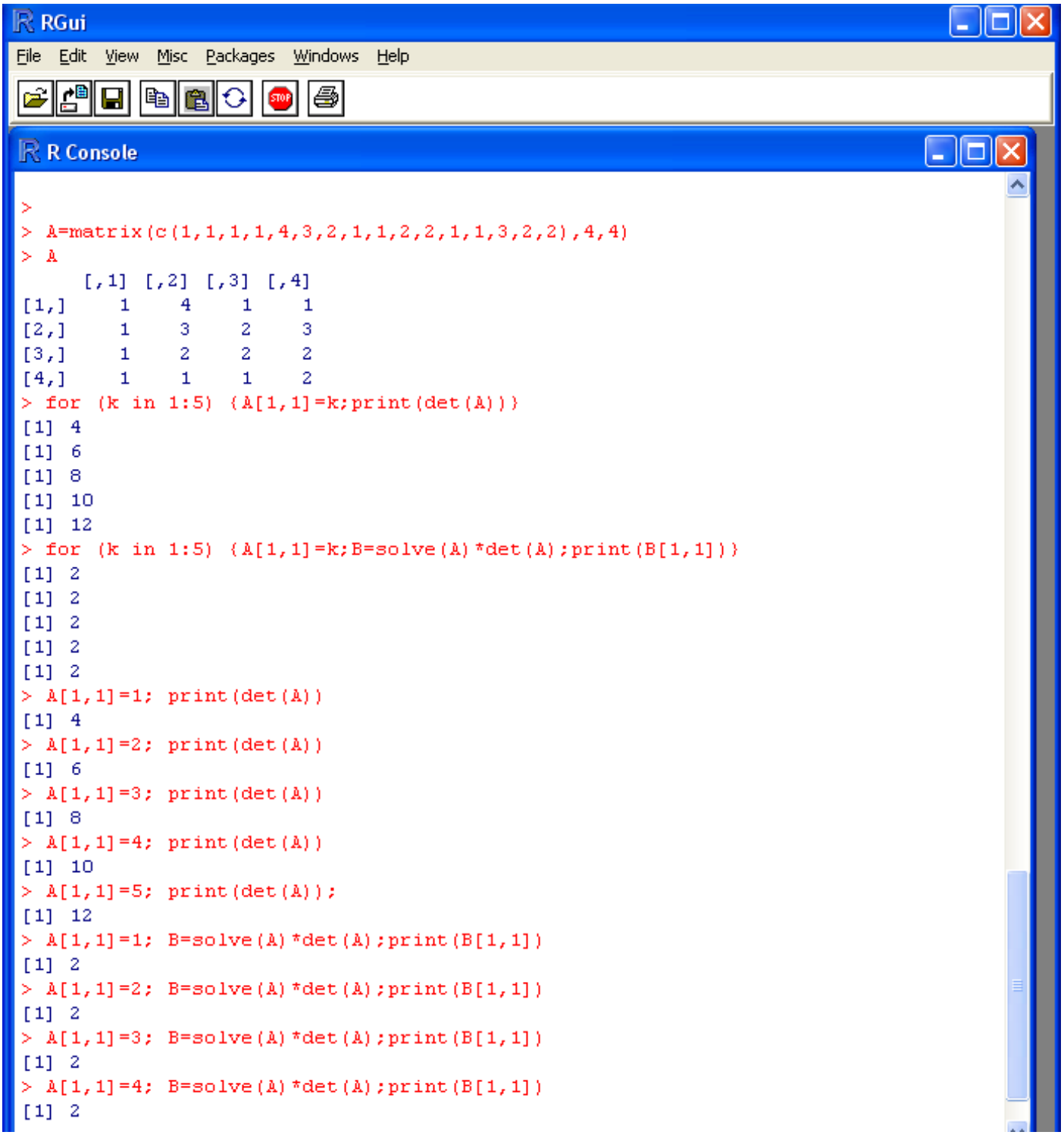
Mathematica



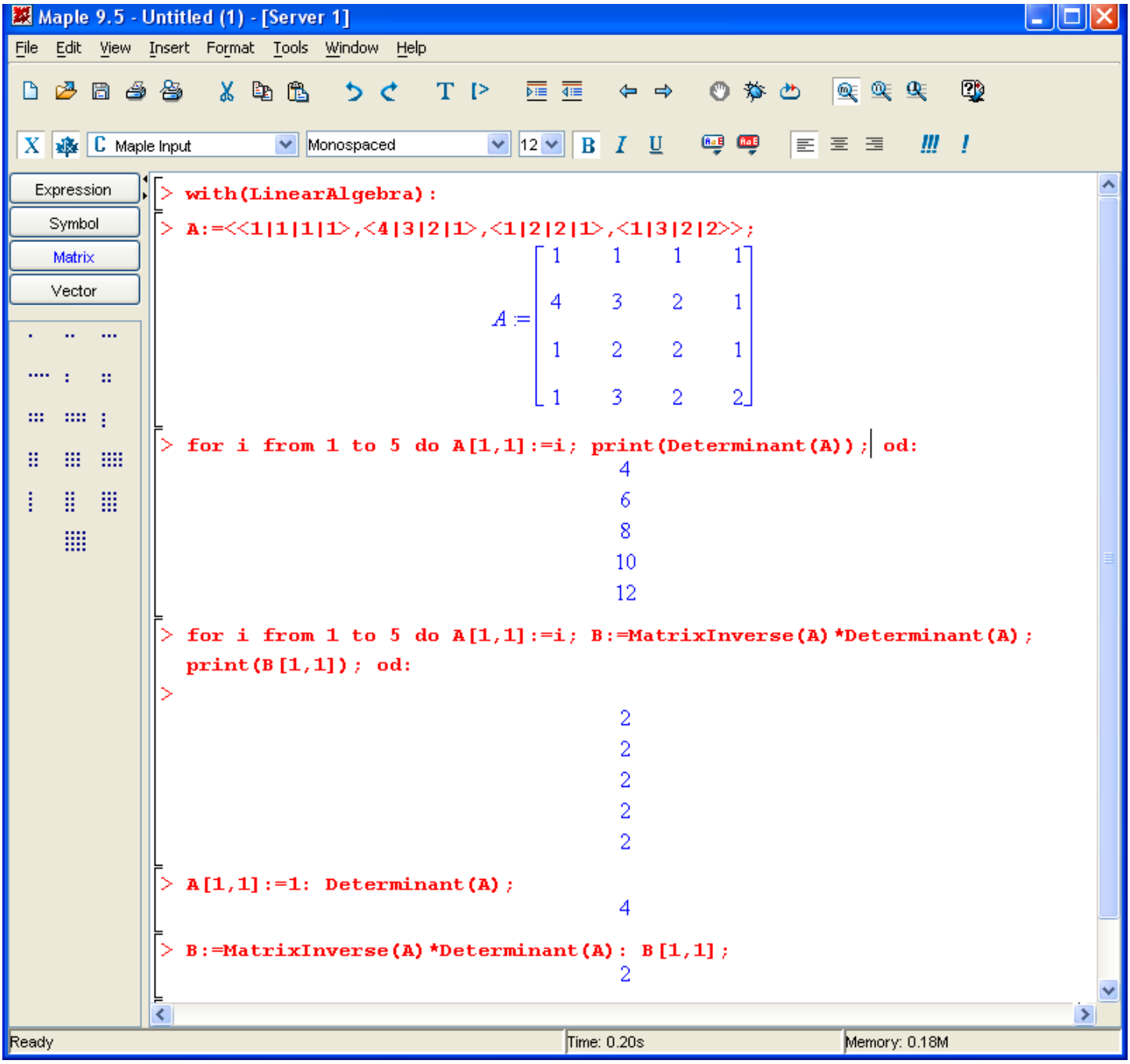
Pylab

```
In [93]: A=array([[1,1,1,1],[4,3,2,1],[1,2,2,1],[1,3,2,2]])

In [94]: for i in range(0,5):
   ....:      A[0,0]=i+1; print(det(A))
   ....:
4.0
6.0
8.0
10.0
12.0

In [95]: for i in range(0,5):
   ....:     A[0,0]=i+1; B=inv(A)*det(A); print(B[0,0])
   ....:
   ....:
2.0
2.0
2.0
2.0
2.0

In [96]: A[0,0]=1; print(det(A))
4.0

In [97]: A[0,0]=2; print(det(A))
6.0

In [98]: A[0,0]=3; print(det(A))
8.0

In [99]: A[0,0]=4; print(det(A))
10.0

In [100]: A[0,0]=5; print(det(A))
12.0

In [101]: A[0,0]=1; B=inv(A)*det(A); print(B[0,0])
2.0

In [102]: A[0,0]=2; B=inv(A)*det(A); print(B[0,0])
2.0

In [103]: A[0,0]=3; B=inv(A)*det(A); print(B[0,0])
2.0

In [104]: A[0,0]=4; B=inv(A)*det(A); print(B[0,0])
2.0

In [105]: A[0,0]=5; B=inv(A)*det(A); print(B[0,0])
2.0
```

Note that matrix(c(1,1,1,1,4,3,2,1,1,2,2,1,1,3,2,1),4,4, byrow=TRUE) gives the matrix “transposed” as in the other languages




```
>
> A=matrix(c(1,1,1,1,4,3,2,1,1,2,2,1,1,3,2,2),4,4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    4    1    1
[2,]    1    3    2    3
[3,]    1    2    2    2
[4,]    1    1    1    2
> for (k in 1:5) {A[1,1]=k;print(det(A))}
[1] 4
[1] 6
[1] 8
[1] 10
[1] 12
> for (k in 1:5) {A[1,1]=k;B=solve(A)*det(A);print(B[1,1])}
[1] 2
[1] 2
[1] 2
[1] 2
[1] 2
> A[1,1]=1; print(det(A))
[1] 4
> A[1,1]=2; print(det(A))
[1] 6
> A[1,1]=3; print(det(A))
[1] 8
> A[1,1]=4; print(det(A))
[1] 10
> A[1,1]=5; print(det(A));
[1] 12
> A[1,1]=1; B=solve(A)*det(A);print(B[1,1])
[1] 2
> A[1,1]=2; B=solve(A)*det(A);print(B[1,1])
[1] 2
> A[1,1]=3; B=solve(A)*det(A);print(B[1,1])
[1] 2
> A[1,1]=4; B=solve(A)*det(A);print(B[1,1])
[1] 2
```


Maple 9.5 - Untitled (1) - [Server 1]
File Edit View Insert Format Tools Window Help
Maple Input
Monospaced
12
Expression
Symbol
Matrix
Vector
> with(LinearAlgebra):
> A:=<<1|1|1|1>,<4|3|2|1>,<1|2|2|1>,<1|3|2|2>>;
A :=
1 1 1 1
4 3 2 1
1 2 2 1
1 3 2 2
> for i from 1 to 5 do A[1,1]:=i; print(Determinant(A)); od:
4
6
8
10
12
> for i from 1 to 5 do A[1,1]:=i; B:=MatrixInverse(A)*Determinant(A);
print(B[1,1]); od:
>
2
2
2
2
2
> A[1,1]:=1: Determinant(A);
4
> B:=MatrixInverse(A)*Determinant(A): B[1,1];
2
Ready
Time: 0.20s
Memory: 0.18M