

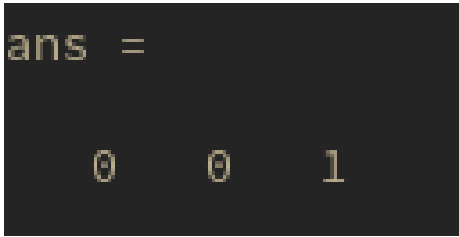
Deber 1

Roberto Alvarado

4 de septiembre del 2021

0.1 Preguntal

```
1 #Initialize the matrix
2 A = [1 1 0; 0 2 3; 3 3 1]
3 #a) Matrix Arithmetics
4 A(1,:) * (-3) + A(3,:)
5 #b) Interchange 3 column with 1
6 C = A(1,:);
7 A(1,:)=A(3,:);
8 A(3,:)=C;
9 #c) Matrix initialization
10 M = [A(:,1) A(:,3)]
11 #d)
12 N = [A(1,:);A(3,:)]
13
14 #e) identity matrix this A
15 I3 = eye(3);
16 B = [A I3];
17 #f) echelon form of the matrix B to get the inverse
18 AI = rref(B) (:,4:6)
19 #g) Check that is the inverse
20 A*AI
```

 <pre>A = 1 1 0 0 2 3 3 3 1</pre>	 <pre>ans = 0 0 1</pre>
Matrix A	a)
 <pre>A = 0 1 1 3 2 0 1 3 3</pre>	 <pre>M = 0 1 3 0 1 3</pre>
c)	d)

```
N =
    0    1    1
    1    3    3
```

e)

```
B =
    0    1    1    1    0    0
    3    2    0    0    1    0
    1    3    3    0    0    1
```

f)

```
AI =
-3.0000e+00 -1.1102e-16 1.0000e+00
 4.5000e+00  5.0000e-01 -1.5000e+00
-3.5000e+00 -5.0000e-01  1.5000e+00
```

f)

```
ans =
    1.0000         0         0
         0    1.0000         0
   -0.0000   -0.0000    1.0000
```

g) Podemos ver que es la matriz inversa

0.2 Pregunta2

```
1 #a) Define matrices
2 A = [2 -1 3;4 2 1;3 2 -5]
3 B = [1;2;-4]
4
5 #Method Gauss-jordan to get the row echolon form
6 AB = [A B]
7 ans = rref(AB) (:,4)
8
9 #b)
10 A = [-1 3;2 0;-1 1]
11 B = [-2;3;1]
12
13 AB = [A B]
14 ans = rref(AB)
```

a) En este caso podemos ver que tiene un resultado este sistema de ecuaciones. Con nuestro procedimiento Octave simplemente nos devuelve la matrix identidad junto al vector resultante.

b) El procedimiento no tiene ninguna solucion, es por eso que octave al momento de hacer el procedimiento devuelve la matriz identidad ya que significa que los vectores son linealmente independientes

```
AB =
     2     -1      3      1
     4      2      1      2
     3      2     -5     -4

ans =
     1.0000         0         0    -0.4390
         0     1.0000         0     1.3415
         0         0     1.0000     1.0732
```

a)

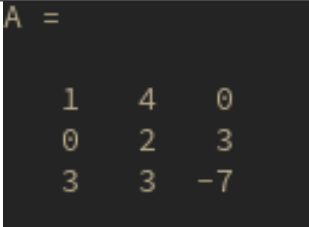
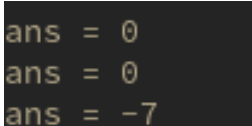
```
AB =
     -1      3     -2
      2      0      3
     -1      1      1

ans =
      1      0      0
      0      1      0
      0      0      1
```

b)

0.3 Pregunta3

```
1 #Initialize de matrix
2 A = [1 4 0;0 2 3;3 3 -7]
3 #a) Get the min for each line , result for each line in vector form
4 for i=1:3
5     min(A(i,:) )
6 end
7
8 #b)Sort in descending order each colum
9 for i=1:3
10    A(:,i) = sort(A(:,i),"descend");
11 endfor
12 A
13
14 #c)Sort in ascending order each line
15 A = [1 4 0;0 2 3;3 3 -7]
16 for i=1:3
17    A(i,:) = sort(A(i,:));
18 endfor
19 A
20
21 #d)List with sorted elements of A in ascending order
22 B =[];
23 for i=1:3
24     for j=1:3
25         B= [B A(i,j) ];
26     end
27 end;
28 sort(B)
29
30 #e) max value of abs(B)
31 max(abs(B))
```

 <pre>A = 1 4 0 0 2 3 3 3 -7</pre>	 <pre>ans = 0 ans = 0 ans = -7</pre>
Matrix A	a)

```
A =
  3  4  3
  1  3  0
  0  2 -7
```

b)

```
A =
  0  1  4
  0  2  3
 -7  3  3
```

c)

```
ans =
 -7  0  0  1  2  3  3  3  4
```

d)

```
ans = 7
```

e)

0.4 Pregunta4

```
1 #Matrix of ones times then mulitply by 7  
2 A = ones(3,5)*7
```

```
A =  
  
    7    7    7    7    7  
    7    7    7    7    7  
    7    7    7    7    7
```

0.5 Pregunta5

```
1 #Random 4x4 real matrix from -5 to 5
2 A = (rand([4,4]) - 0.5)*10
```

```
A =

-4.7526    1.0720   -4.9166    4.1225
-4.5750    2.3192   -1.8269   -0.4952
-1.1847   -0.1763    4.0221    1.9838
-0.7369   -2.6442   -4.7777   -0.9064
```

0.6 Pregunta6

```
1 #Random 4x4 integer matrix from -5 to 5  
2 A=int32((rand([4 4]) -0.5)*10)
```

```
A =  
  
     4     -3     4     2  
    -2      3      3      3  
    -3      4     -3     -3  
     4      3     -5     -5
```


0.7 Pregunta7

```

1 A = magic(5)
2
3 #a
4 B = [A(:,2) A(:,4)]
5 #b
6 C = [A(:,1) A(:,3) A(:,5)]
7 #c
8 d = [A(:,3)]
9 #d
10 A(:,3) = []

```

<pre> A = 17 24 1 8 15 23 5 7 14 16 4 6 13 20 22 10 12 19 21 3 11 18 25 2 9 </pre>	<pre> B = 24 8 5 14 6 20 12 21 18 2 </pre>
<p>Matrix A</p> <pre> C = 17 1 15 23 7 16 4 13 22 10 19 3 11 25 9 </pre>	<p>a)</p> <pre> d = 1 7 13 19 25 </pre>
<p>b)</p> <pre> A = 17 24 8 15 23 5 14 16 4 6 20 22 10 12 21 3 11 18 2 9 </pre>	<p>c)</p>
<p>d)</p>	

0.8 Pregunta8

```
1 #Initilization of vector
2 x = (0:pi/2:2*pi)
3 #Matrix definition
4 A = [x; sin(x); cos(x)]
```

```
x =
    0    1.5708    3.1416    4.7124    6.2832

A =
    0    1.5708    3.1416    4.7124    6.2832
    0    1.0000    0.0000   -1.0000   -0.0000
    1.0000    0.0000   -1.0000   -0.0000    1.0000
```

0.9 Pregunta9

```
1 #Vectors
2 a = [1 3 5 7]
3 b = [2 4 6 8]
4 # AT * B = Expected Matrix
5 a'*b
```

```
a =
     1     3     5     7

b =
     2     4     6     8

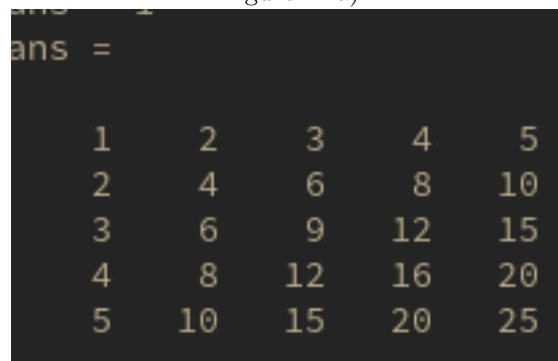
ans =

     2     4     6     8
     6    12    18    24
    10    20    30    40
    14    28    42    56
```

0.10 Preguntal0

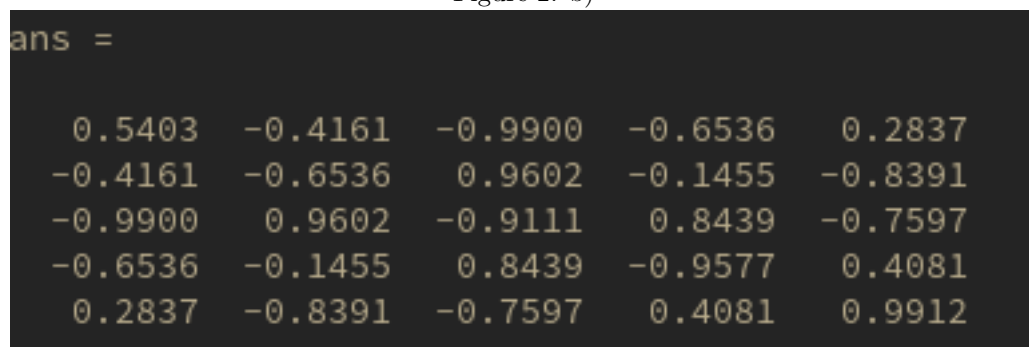
```
1 1,
2 function matrix = getMatrixA(na)
3     matrix = zeros(na);
4     for i=1:na
5         for j=1:na
6             matrix(i,j)=i*j;
7         endfor
8     endfor
9 endfunction
10
11 #try with a 5x5
12 getMatrixA(5)
13
14 #Function where c_ij = cos(ij)
15 2,
16 function matrix = getMatrixB(na)
17     matrix = zeros(na);
18     for i=1:na
19         for j=1:na
20             matrix(i,j)=cos(i*j);
21         endfor
22     endfor
23 endfunction
24
25 #try with a 5x5
26 getMatrixB(5)
```

Figure 1: a)



1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Figure 2: b)



0.5403	-0.4161	-0.9900	-0.6536	0.2837
-0.4161	-0.6536	0.9602	-0.1455	-0.8391
-0.9900	0.9602	-0.9111	0.8439	-0.7597
-0.6536	-0.1455	0.8439	-0.9577	0.4081
0.2837	-0.8391	-0.7597	0.4081	0.9912

0.11 Preguntal1

```
1 #Vector of vectos
2 a = [1 1 1 1]
3 A = [a;a*2;a*3;a*4]
4
5 #Change each vector
6 A = ones(4,4);
7 A(2,:) = A(2,:) * 2;
8 A(3,:) = A(3,:) * 3;
9 A(4,:) = A(4,:) * 4;
10 A
11
12 #change each element
13 A = zeros(4,4);
14 for i=1:4
15     for j=1:4
16         A(i,j)=i;
17     end
18 end
19 A
```

```
A =
     1     1     1     1
     2     2     2     2
     3     3     3     3
     4     4     4     4
```

```
A =
     1     1     1     1
     2     2     2     2
     3     3     3     3
     4     4     4     4
```

```
A =
     1     1     1     1
     2     2     2     2
     3     3     3     3
     4     4     4     4
```

0.12 Pregunta12

```
1 A = magic(5)
2 #diagonal to 0
3 A(find(eye(5)))= 0
```

```
A =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

A =
     0    24     1     8    15
    23     0     7    14    16
     4     6     0    20    22
    10    12    19     0     3
    11    18    25     2     0
```

0.13 Pregunta13

```
1 #diagonal to any vector
2 A = magic(5)
3 A(find(eye(5)))= [1 2 3 4 5]
```

```
A =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

A =
     1    24     1     8    15
    23     2     7    14    16
     4     6     3    20    22
    10    12    19     4     3
    11    18    25     2     5
```

0.14 Pregunta14

```
1 #reshape
2 A = magic(6)
3 A = reshape(A,1,36);
4 A = sort(A);
5 A = reshape(A,6,6)
```

```
A =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

A =
     1     7    13    19    25    31
     2     8    14    20    26    32
     3     9    15    21    27    33
     4    10    16    22    28    34
     5    11    17    23    29    35
     6    12    18    24    30    36
```


0.15 Pregunta15

```
1 #Matrix
2 A = randn([4,4])*10
3 #We find the cells where there are negative numbers and then change them
4 A(find(A<0))=0
```

```
A =  
  
    13.9451    0.3944    12.6413    4.3256  
    1.6931    2.6259    0.8924   -1.8758  
   -14.6516  -13.8151  -13.1674   -1.1379  
    1.2574   -4.1305   -6.9840    5.8196  
  
A =  
  
    13.9451    0.3944    12.6413    4.3256  
    1.6931    2.6259    0.8924         0  
         0         0         0         0  
    1.2574         0         0    5.8196
```

0.16 Pregunta16

```
1 A = (rand([5,5]) - 0.5) * 10;  
2 A(find(A > 4)) = A(find(A > 4)) - 1;  
3 A(find(A < -4)) = A(find(A < -4)) + 1  
4 A(find(abs(A) < 1)) = 0
```

```
A =  
2.502694 -2.258693 2.552242 -1.188064 0.828395  
-0.202916 -2.082893 -2.344051 1.658046 -3.696652  
-3.653038 -1.043305 -0.385198 -3.787259 -1.699279  
-0.420395 -3.654172 -0.022120 3.432786 -2.916493  
-2.786579 3.837201 3.736741 -3.828031 2.684099  
  
A =  
2.5027 -2.2587 2.5522 -1.1881 0  
0 -2.0829 -2.3441 1.6580 -3.6967  
-3.6530 -1.0433 0 -3.7873 -1.6993  
0 -3.6542 0 3.4328 -2.9165  
-2.7866 3.8372 3.7367 -3.8280 2.6841
```

0.17 Pregunta17

```
1 A= magic(randi(10))
2 #1
3 length(A(1,:))*length(A)
4 #2
5 prod(size(A))
6 #3
7 length(vec(A))
```

```
A =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

ans = 25
ans = 25
ans = 25
```

0.18 Pregunta18

```
1 disp("A = rand([4,6])\nA = A + 2")
2 A = rand([4,6])
3 A = A + 2
```

```
A =
    9.1546e-01    8.6243e-01    7.2887e-01    2.7630e-02    6.2726e-01    8.6600e-01
    9.5689e-01    5.8658e-01    3.5084e-02    5.4818e-01    2.6285e-01    2.2204e-01
    2.6278e-01    8.5469e-04    5.5362e-01    2.7040e-01    4.0244e-01    4.0901e-01
    5.1824e-01    7.9622e-01    7.0174e-01    3.2465e-01    1.9422e-01    8.9957e-01

A =
    2.9155    2.8624    2.7289    2.0276    2.6273    2.8660
    2.9569    2.5866    2.0351    2.5482    2.2628    2.2220
    2.2628    2.0009    2.5536    2.2704    2.4024    2.4090
    2.5182    2.7962    2.7017    2.3247    2.1942    2.8996
```

0.19 Pregunta19

```

1 #a
2 a = vec([1:101])
3 #b
4 b = vec([6:-0.5:3])
5 #c
6 c = vec([5.3:0.7:9.2])
7 #d
8 d = randi(10,1,45)

```

<pre>ans = 1 2 3 4 5 6 7 8 9 10</pre>	<pre>b = 6.0000 5.5000 5.0000 4.5000 4.0000 3.5000 3.0000</pre>
<pre>M = 0 1 3 0 1 3</pre>	<pre>d = Columns 1 through 16: 2 3 2 3 3 8 7 7 9 7 5 10 9 6 6 4 Columns 17 through 32: 1 7 9 2 10 3 6 4 10 1 6 6 3 1 10 7 Columns 33 through 45: 8 5 1 10 10 9 10 5 4 8 4 9 10</pre>