Lab 1: Introduction to MATLAB: Basic Commands and Array Manipulation

Name	**DANISH ZULFIQAR**
Registration Number	**FA21-BEE-042**
Class/section	**4-A**
Instructor’s Name	**DR. BILAL TARIQ**

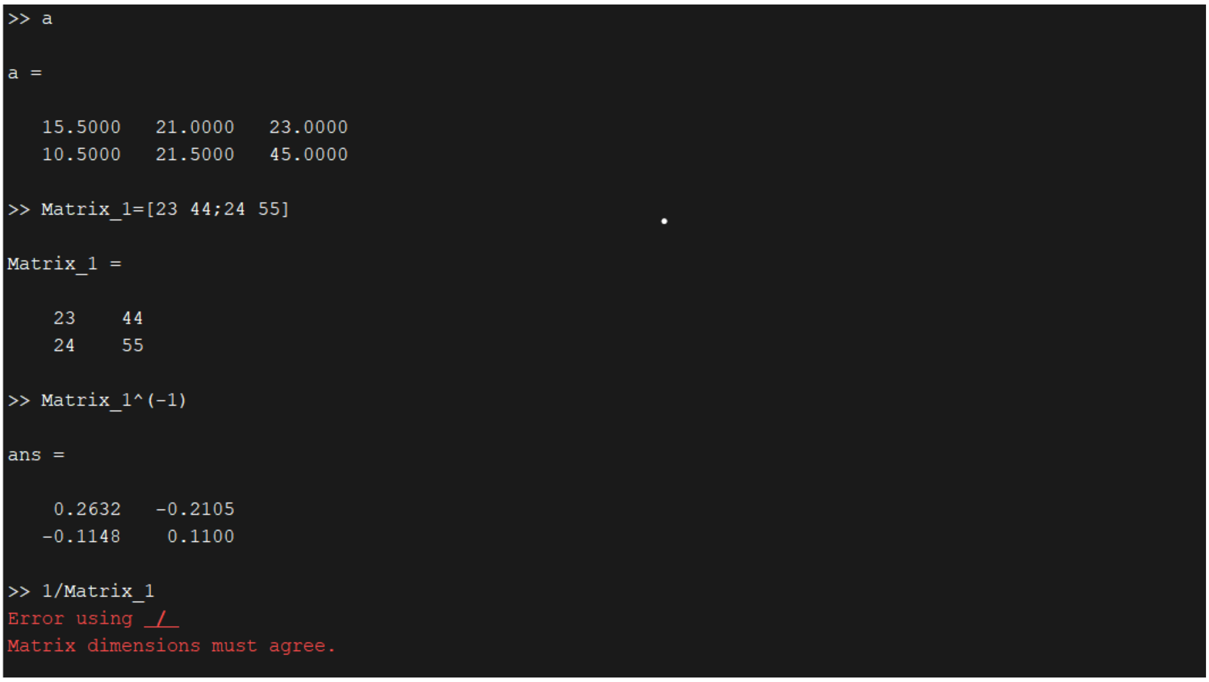
Objectives:

Once MATLAB is installed; it is time to start using it. MATLAB is best understood by entering the instructions in your computer one at a time and by observing and studying the responses. Learning by doing is probably the most effective way to maximize retention.

In Lab Tasks:

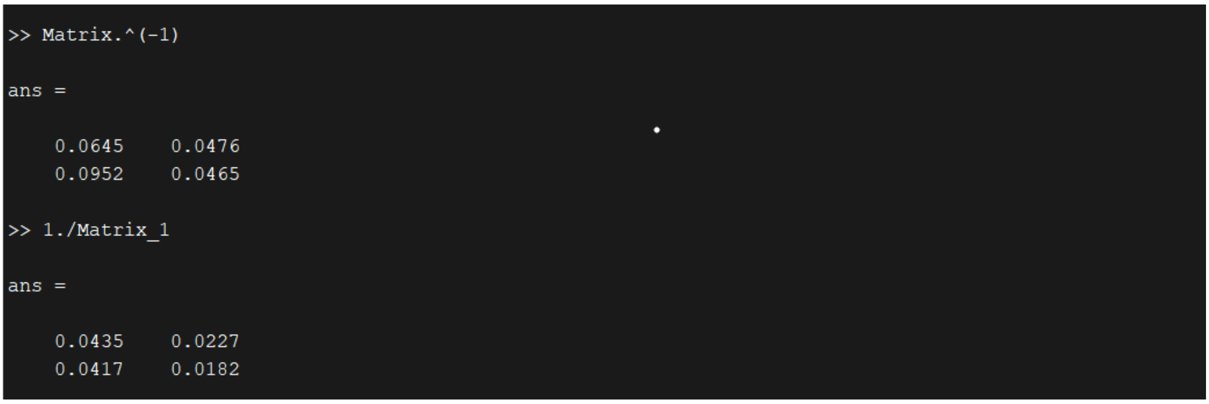
Task 1:

a) False



```
>> a

a =

   15.5000   21.0000   23.0000
   10.5000   21.5000   45.0000

>> Matrix_1=[23 44;24 55]

Matrix_1 =

    23    44
    24    55

>> Matrix_1^(-1)

ans =

    0.2632   -0.2105
   -0.1148    0.1100

>> 1/Matrix_1
Error using  / 
Matrix dimensions must agree.
```

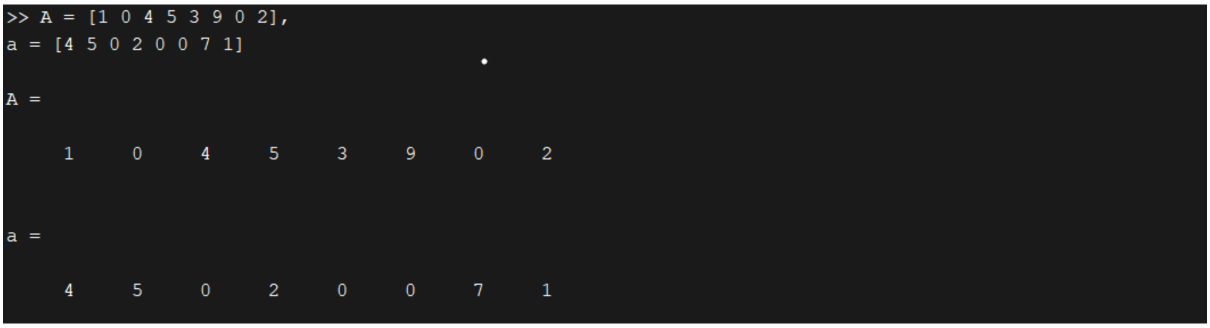
b) False



```
>> Matrix.^(-1)

ans =

    0.0645    0.0476
    0.0952    0.0465

>> 1./Matrix_1

ans =

    0.0435    0.0227
    0.0417    0.0182
```

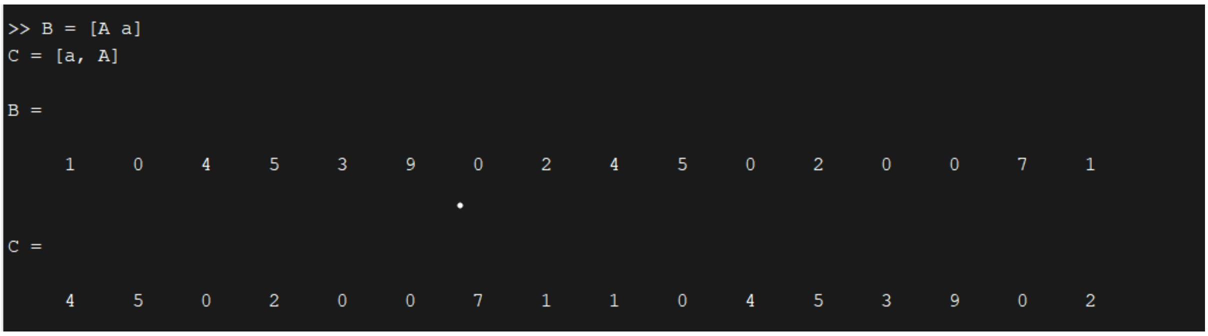
Task 2:

a)



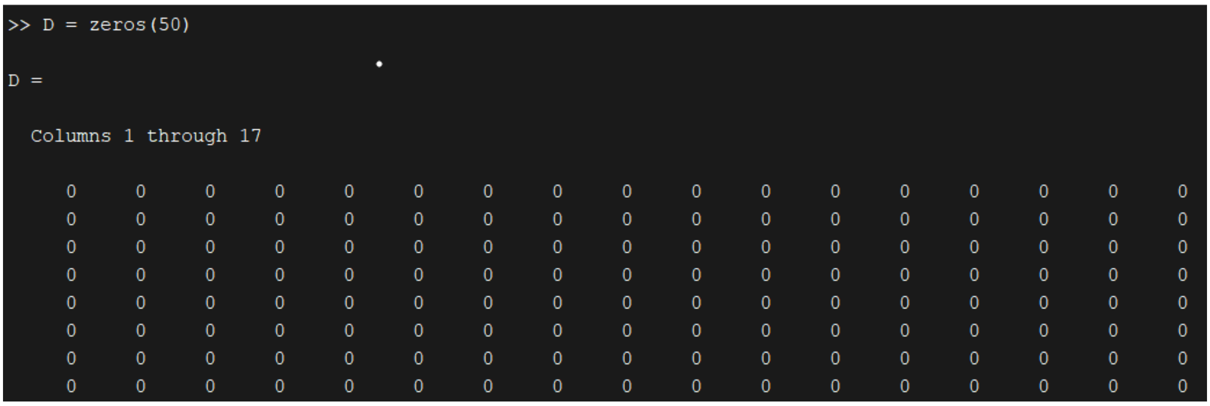
```
>> A = [1 0 4 5 3 9 0 2],
a = [4 5 0 2 0 0 7 1]

A =

     1     0     4     5     3     9     0     2


a =

     4     5     0     2     0     0     7     1
```

b)

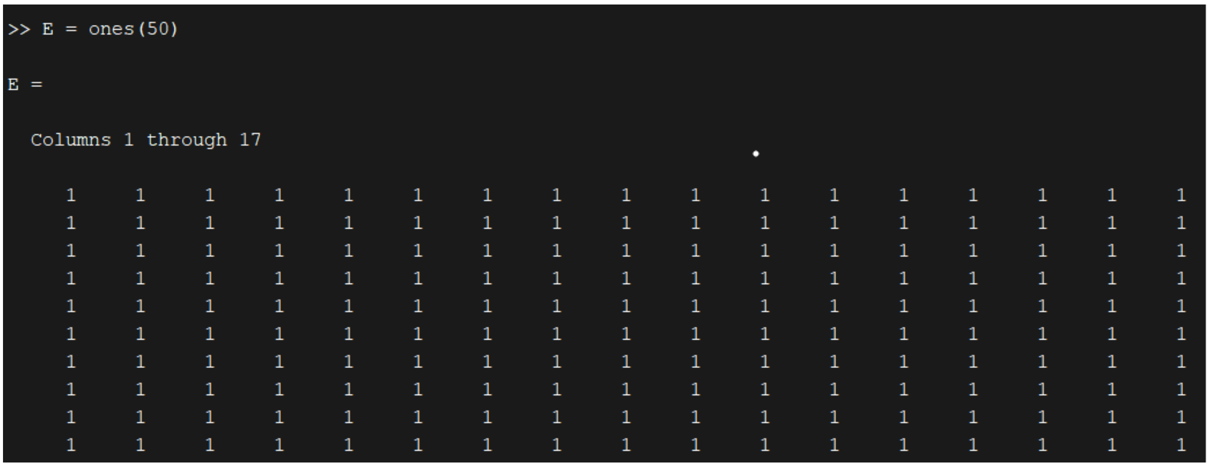


```
>> B = [A a]
C = [a, A]

B =

     1     0     4     5     3     9     0     2     4     5     0     2     0     0     7     1


C =

     4     5     0     2     0     0     7     1     1     0     4     5     3     9     0     2
```

c)

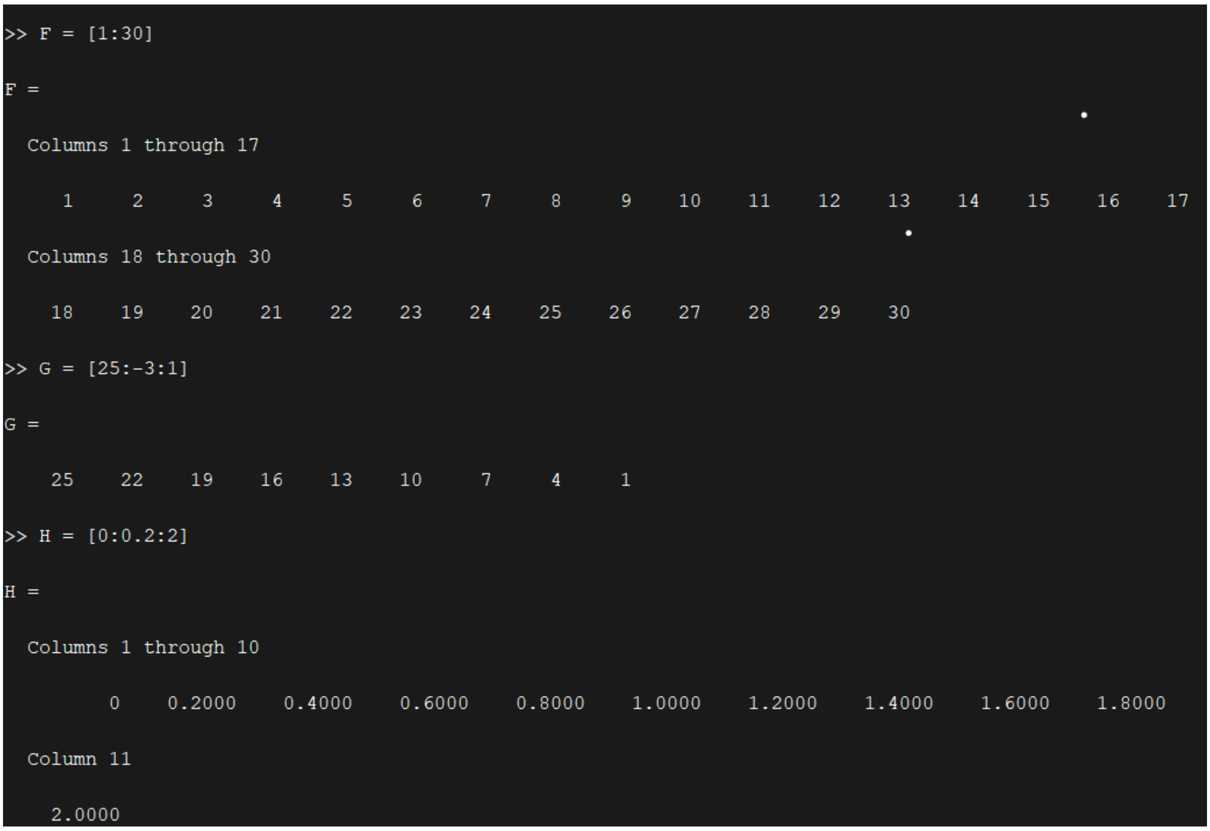


```
>> D = zeros(50)

D =

  Columns 1 through 17

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```



```
>> E = ones(50)

E =

  Columns 1 through 17

     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
```

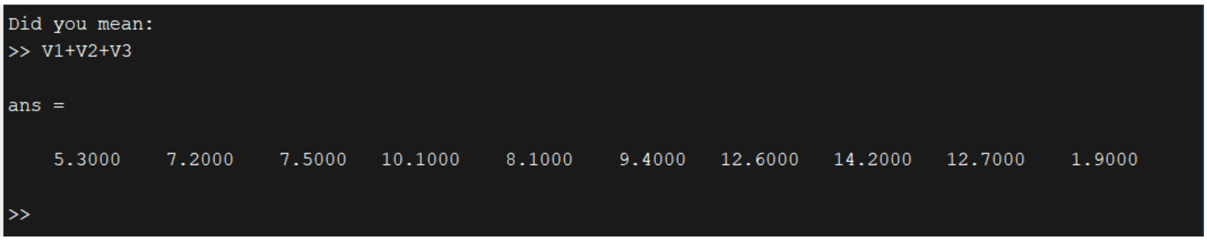
d)



```
>> F = [1:30]

F =

  Columns 1 through 17

     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17

  Columns 18 through 30

    18    19    20    21    22    23    24    25    26    27    28    29    30

>> G = [25:-3:1]

G =

    25    22    19    16    13    10     7     4     1

>> H = [0:0.2:2]

H =

  Columns 1 through 10

         0    0.2000    0.4000    0.6000    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000

  Column 11

    2.0000
```

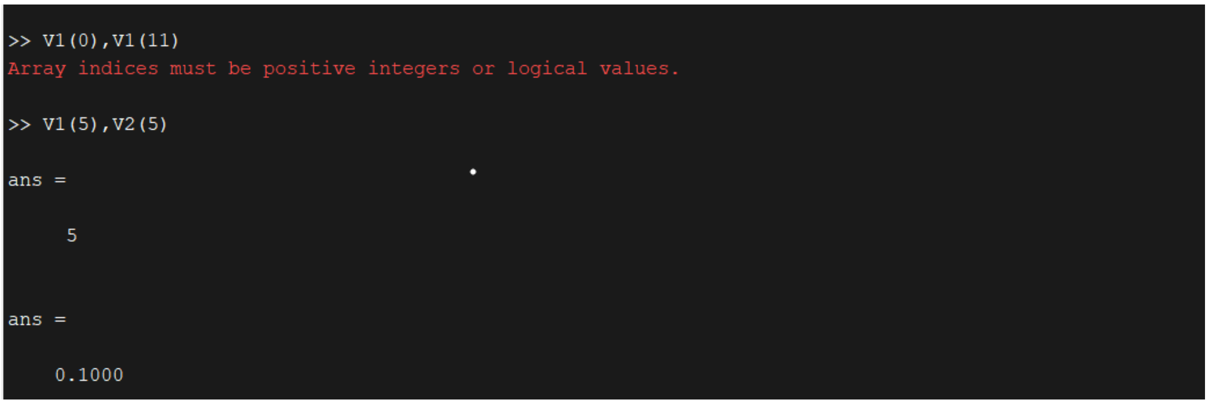
Task 3:

a)



```
Did you mean:
>> V1+V2+V3

ans =

    5.3000    7.2000    7.5000   10.1000    8.1000    9.4000   12.6000   14.2000   12.7000    1.9000

>>
```

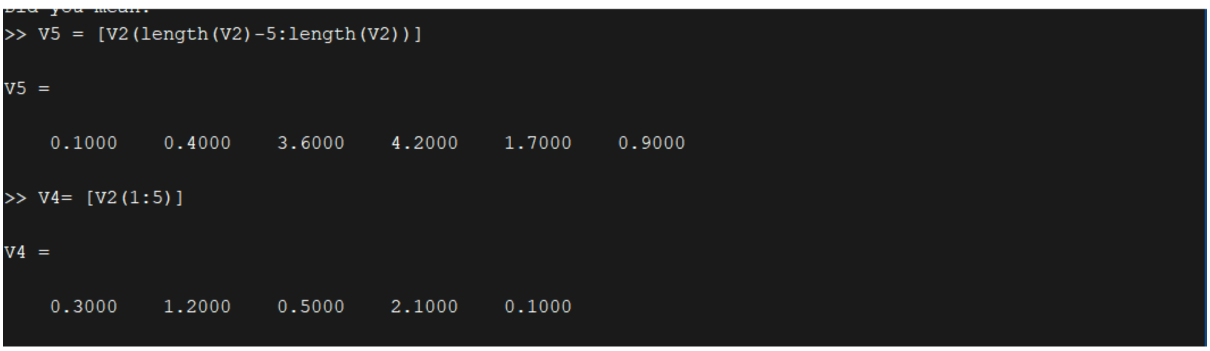
b)



```
>> V1(0),V1(11)
Array indices must be positive integers or logical values.

>> V1(5),V2(5)

ans =

     5


ans =

    0.1000
```

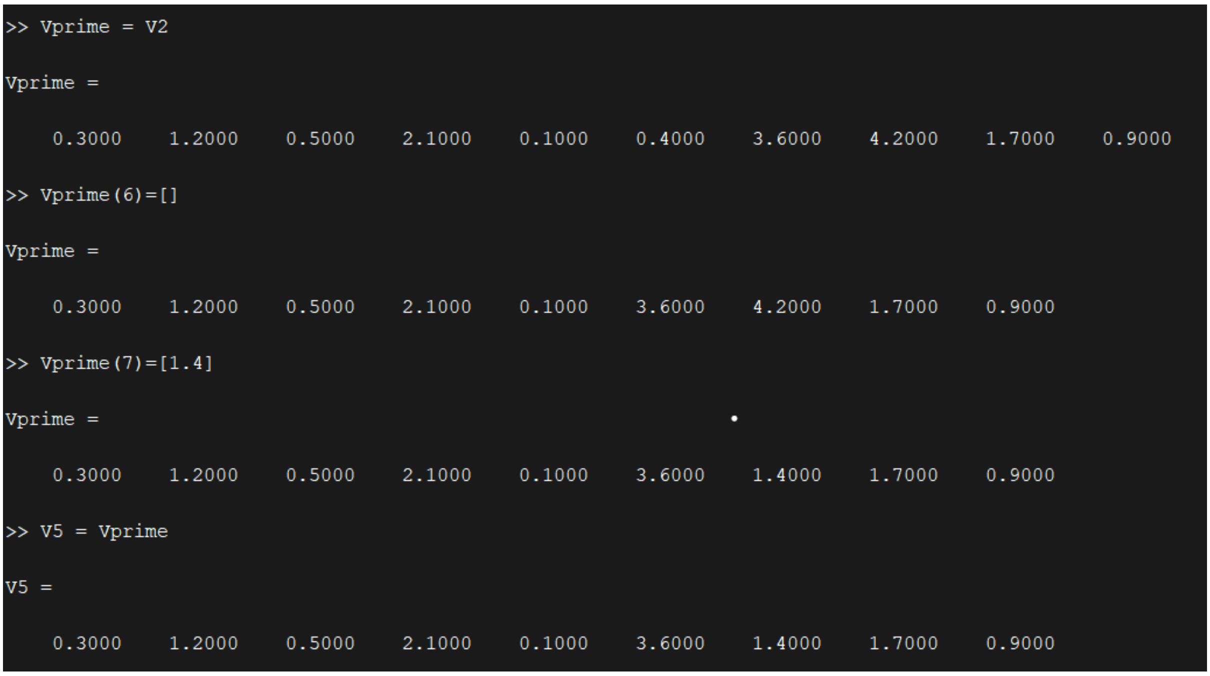
As vectors are formed using arrays in Matlab there indexing starts from 1 to the number or columns. Zero index and column+1 index does not exist.

c)

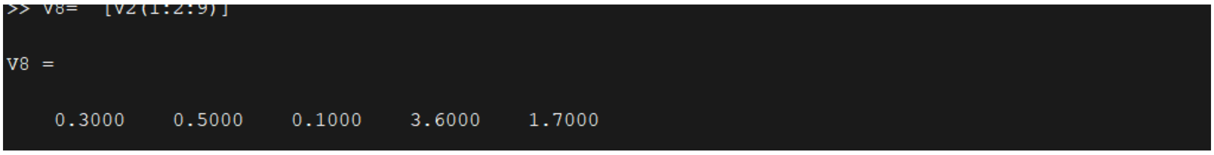


```
Did you mean:
>> V5 = [V2(length(V2)-5:length(V2))]

V5 =

    0.1000    0.4000    3.6000    4.2000    1.7000    0.9000

>> V4= [V2(1:5)]

V4 =

    0.3000    1.2000    0.5000    2.1000    0.1000
```

d)

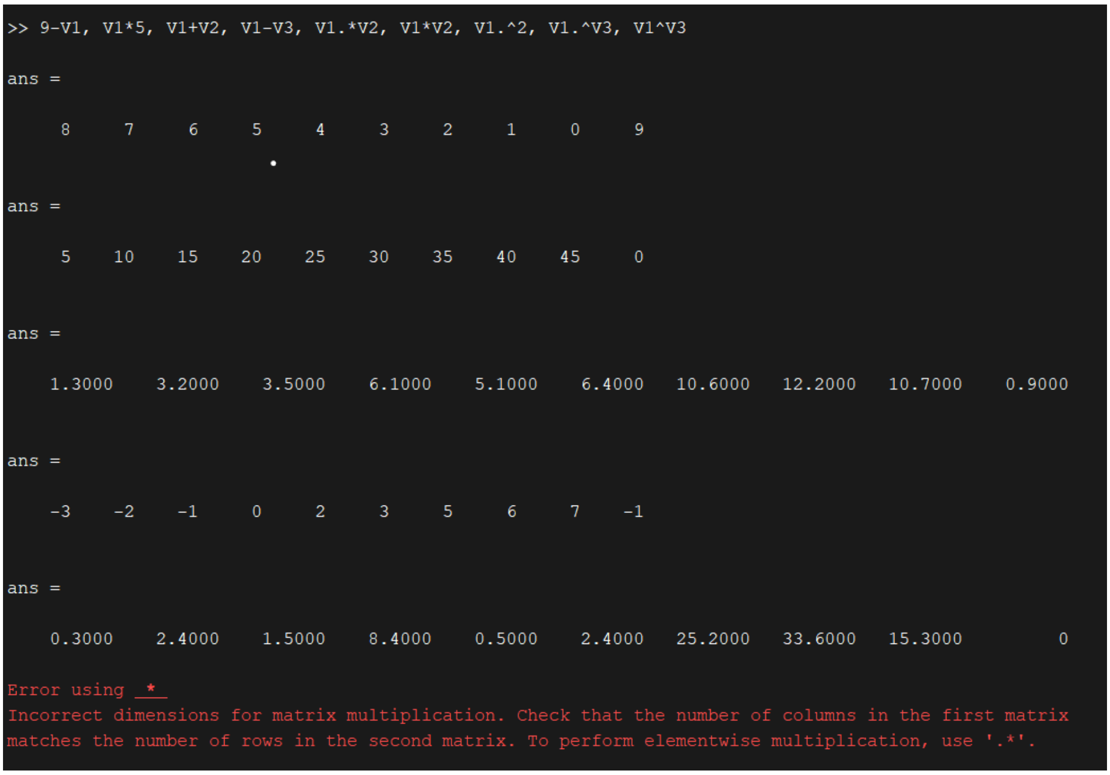


```
>> Vprime = V2

Vprime =

    0.3000    1.2000    0.5000    2.1000    0.1000    0.4000    3.6000    4.2000    1.7000    0.9000

>> Vprime(6)=[]

Vprime =

    0.3000    1.2000    0.5000    2.1000    0.1000    3.6000    4.2000    1.7000    0.9000

>> Vprime(7)=[1.4]

Vprime =

    0.3000    1.2000    0.5000    2.1000    0.1000    3.6000    1.4000    1.7000    0.9000

>> V5 = Vprime

V5 =

    0.3000    1.2000    0.5000    2.1000    0.1000    3.6000    1.4000    1.7000    0.9000
```



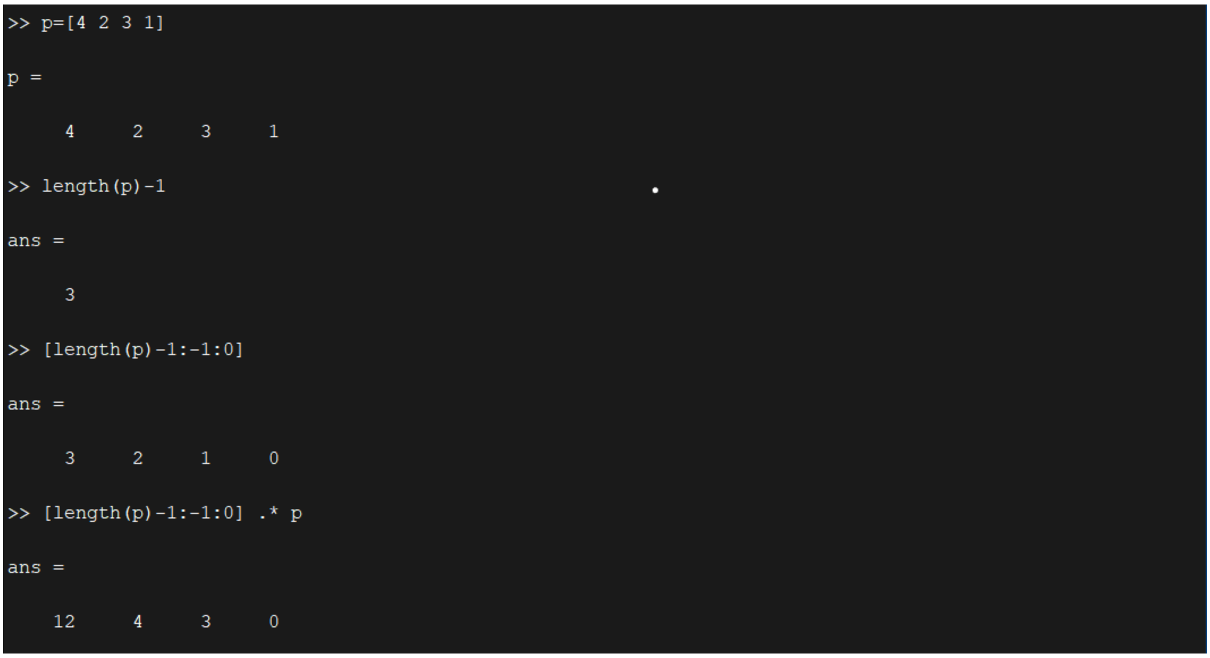
```
>> V8=  [V2(1:2:9)]

V8 =

    0.3000    0.5000    0.1000    3.6000    1.7000
```

e)



```
>> 9-V1, V1*5, V1+V2, V1-V3, V1.*V2, V1*V2, V1.^2, V1.^V3, V1^V3

ans =

     8     7     6     5     4     3     2     1     0     9


ans =

     5    10    15    20    25    30    35    40    45     0


ans =

    1.3000    3.2000    3.5000    6.1000    5.1000    6.4000   10.6000   12.2000   10.7000    0.9000


ans =

    -3    -2    -1     0     2     3     5     6     7    -1


ans =

    0.3000    2.4000    1.5000    8.4000    0.5000    2.4000   25.2000   33.6000   15.3000         0

Error using  * 
Incorrect dimensions for matrix multiplication. Check that the number of columns in the first matrix
matches the number of rows in the second matrix. To perform elementwise multiplication, use '.*'.
```

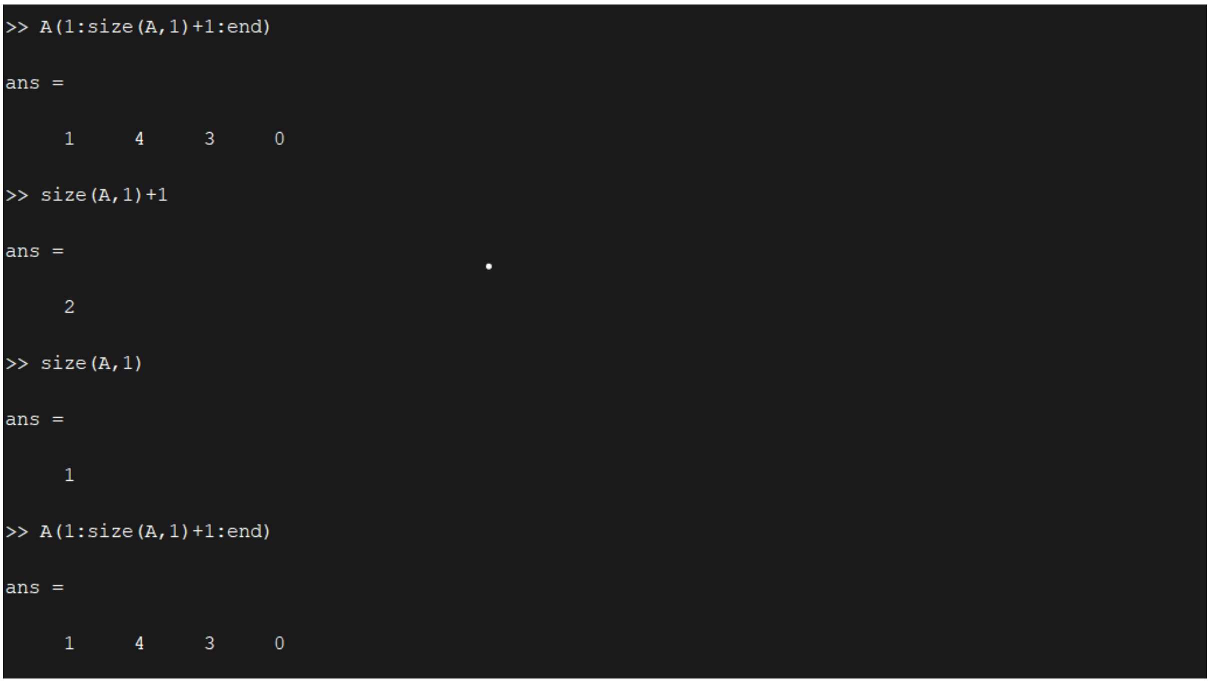
Task 4:



```
>> p=[4 2 3 1]

p =

     4     2     3     1

>> length(p)-1

ans =

     3

>> [length(p)-1:-1:0]

ans =

     3     2     1     0

>> [length(p)-1:-1:0] .* p

ans =

    12     4     3     0
```

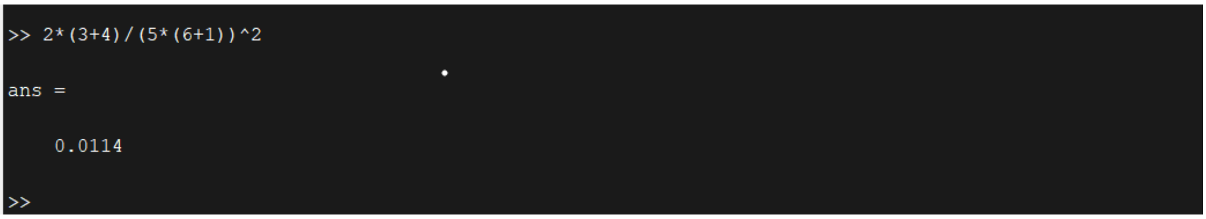
length (p) gives 4 after subtracting 1 it becomes 3, now we are making a vector witch starts from 3 decrementing values up to 0. This vector is then element wise multiplied with the original p vector.

Task 5:



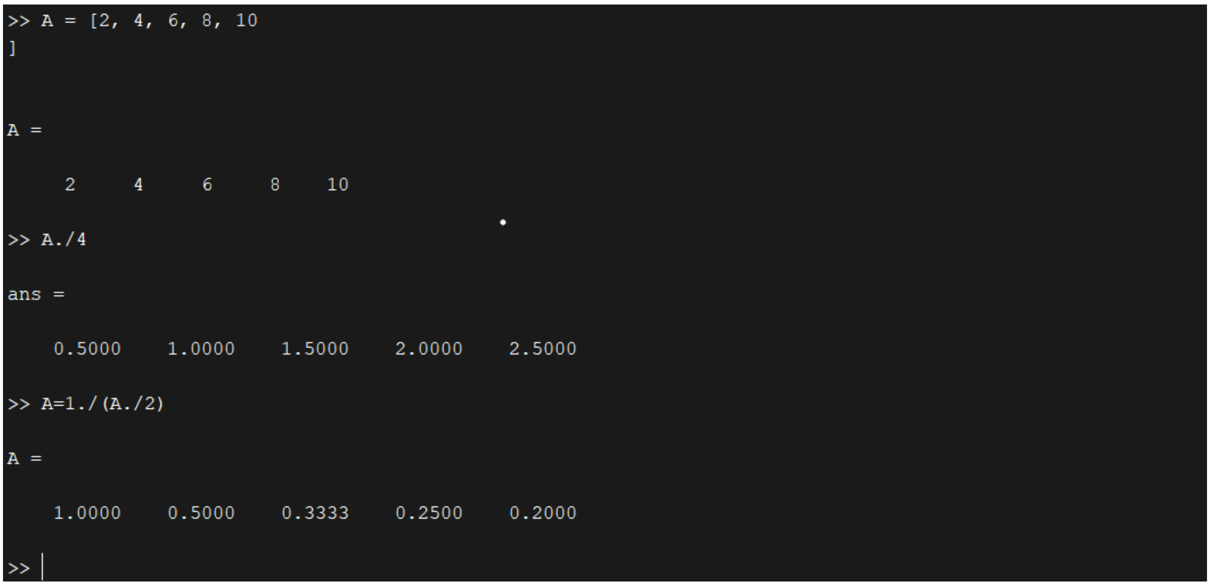
```
>> A(1:size(A,1)+1:end)

ans =

     1     4     3     0

>> size(A,1)+1

ans =

     2

>> size(A,1)

ans =

     1

>> A(1:size(A,1)+1:end)

ans =

     1     4     3     0
```

Task 6:



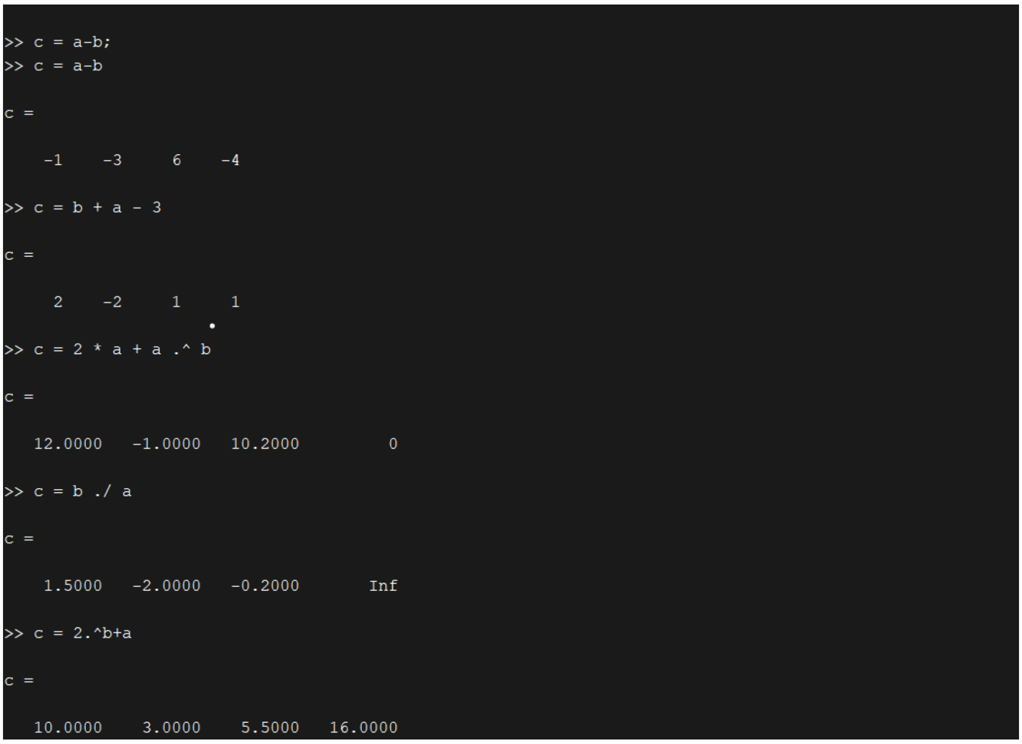
```
>> 2*(3+4)/(5*(6+1))^2

ans =

    0.0114

>>
```

Task 7:

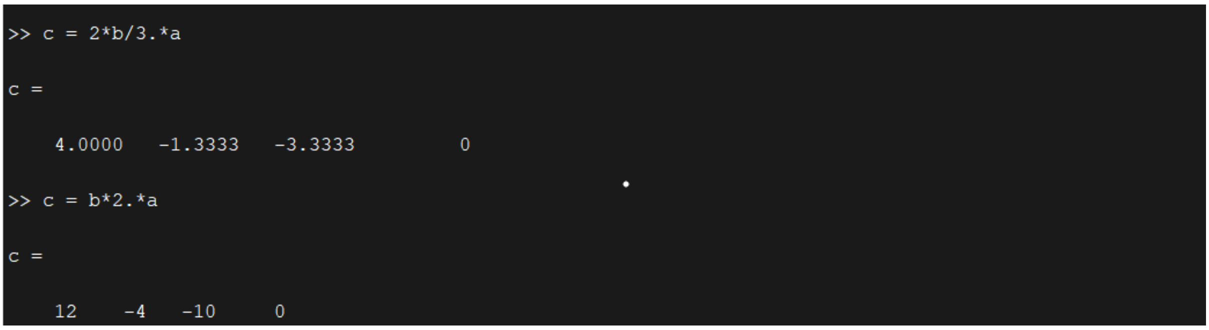


```
>> A = [2, 4, 6, 8, 10
]


A =

     2     4     6     8    10

>> A./4

ans =

    0.5000    1.0000    1.5000    2.0000    2.5000

>> A=1./(A./2)

A =

    1.0000    0.5000    0.3333    0.2500    0.2000

>> 
```

Task 8:

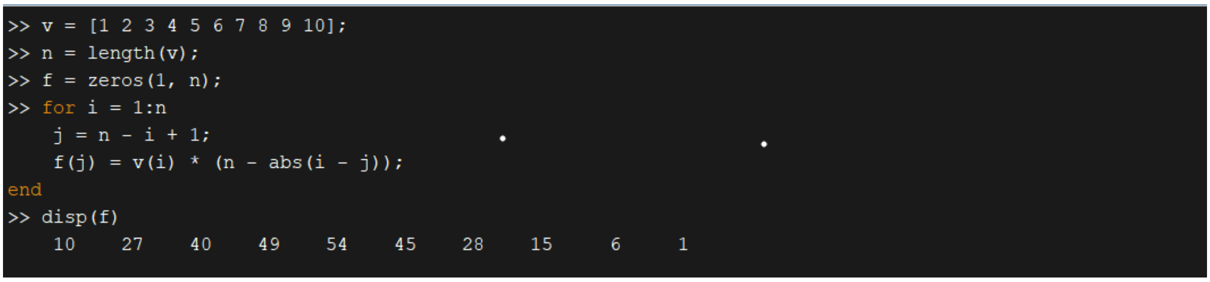


```
>> c = a-b;
>> c = a-b

c =

    -1    -3     6    -4

>> c = b + a - 3

c =

     2    -2     1     1

>> c = 2 * a + a .^ b

c =

   12.0000   -1.0000   10.2000         0

>> c = b ./ a

c =

    1.5000   -2.0000   -0.2000       Inf

>> c = 2.^b+a

c =

   10.0000    3.0000    5.5000   16.0000
```



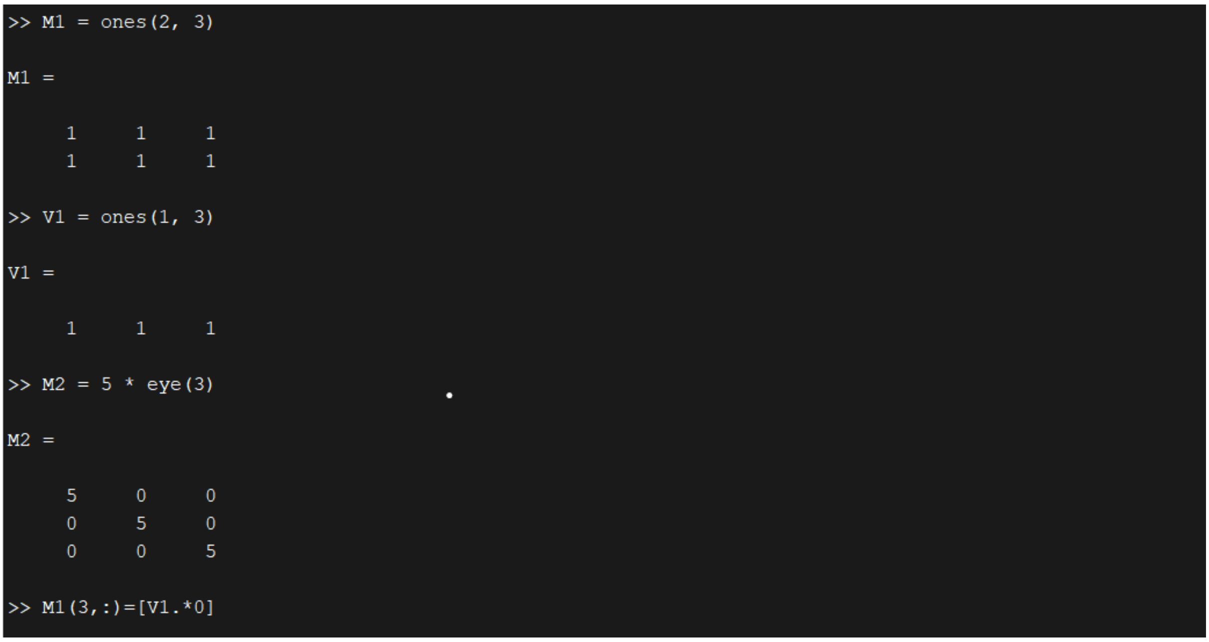
```
>> c = 2*b/3.*a

c =

    4.0000   -1.3333   -3.3333         0

>> c = b*2.*a

c =

    12    -4   -10     0
```

Task 9:



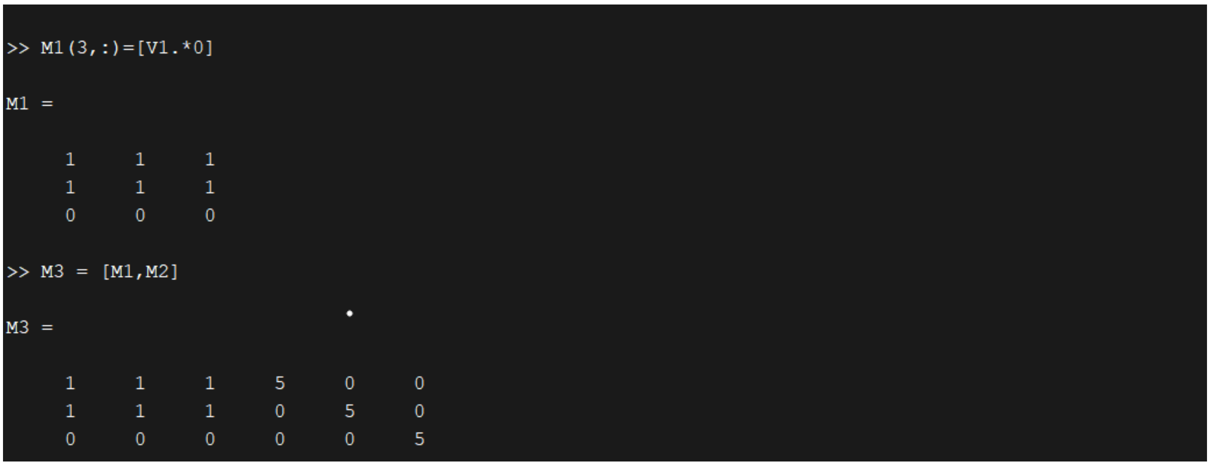
```
>> v = [1 2 3 4 5 6 7 8 9 10];
>> n = length(v);
>> f = zeros(1, n);
>> for i = 1:n
    j = n - i + 1;
    f(j) = v(i) * (n - abs(i - j));
end
>> disp(f)
    10    27    40    49    54    45    28    15     6     1
```

Task 10:



```
>> M1 = ones(2, 3)

M1 =

     1     1     1
     1     1     1

>> V1 = ones(1, 3)

V1 =

     1     1     1

>> M2 = 5 * eye(3)

M2 =

     5     0     0
     0     5     0
     0     0     5

>> M1(3,:)=[V1.*0]
```

d)



```
>> M1(3,:)=[V1.*0]

M1 =

     1     1     1
     1     1     1
     0     0     0

>> M3 = [M1,M2]

M3 =

     1     1     1     5     0     0
     1     1     1     0     5     0
     0     0     0     0     0     5
```

Post Lab Tasks:

Critical Analysis/Conclusion:

In this lab I learnt the basics of Matlab. I learnt how Matlab uses arrays to make vectors and matrices from vectors. I used the basic concepts of linear algebra to visualize the effects of different operations performed on vectors and matrices. I also had hands on experience with different functions on Matlab.