Roll. No. A016	Name: Varun Khadayate
Class B.Tech CsBs	Batch: 1
Date of Experiment: 29-07-2021	Subject: IT/WS

1. If a variable has the dimensions 3 x 4, could it be considered to be (Check all that apply):
   a. a matrix
   b. a row vector
   c. a column vector
   d. a scalar

2. If a variable has the dimensions 1 x 5, could it be considered to be (Check all that apply):
   a. a matrix
   b. a row vector
   c. a column vector
   d. a scalar

3. If a variable has the dimensions 5 x 1, could it be considered to be (Check all that apply):
   a. a matrix
   b. a row vector
   c. a column vector
   d. a scalar

4. If a variable has the dimensions 1 x 1, could it be considered to be (Check all that apply):
   a. a matrix
   b. a row vector
   c. a column vector
   d. a scalar

5. Using the colon operator, create the following row vectors

   2 3 4 5 6 7

   1.1000 1.3000 1.5000 1.7000

   8 6 4 2

```
>> 2:7

ans =

     2.00     3.00     4.00     5.00     6.00     7.00
>> 1.1000:0.2000:1.7000

ans =

  1.1000  1.3000  1.5000  1.7000
>> 8:-2:2

ans =

   8   6   4   2
```

6. Using a built-in function, create a vector vec which consists of 20 equally spaced points in the range from –pi to +pi.

```
>> vec = linspace(0,2*pi,20)

vec =

 Columns 1 through 12

      0   0.3307   0.6614   0.9921   1.3228   1.6535   1.9842   2.3149   2.6456   2.9762   3.3069
3.6376

 Columns 13 through 20

   3.9683   4.2990   4.6297   4.9604   5.2911   5.6218   5.9525   6.2832
```

7. Write an expression using linspace that will result in the same as 2: 0.2: 3

```
>> linspace(2,3,6)

ans =

   2.0000   2.2000   2.4000   2.6000   2.8000   3.0000
```

8. Using the colon operator and also the linspace function, create the following row vectors:

-5 -4 -3 -2 -1

5 7 9

8 6 4

```
>> -5:-1

ans =

   -5   -4   -3   -2   -1

>> linspace(-5,-1,5)

ans =

   -5   -4   -3   -2   -1

>> 5:2:9

ans =

    5    7    9

>> linspace(5,9,3)

ans =
```

```
   5    7    9

>> 8:-2:4

ans =

   8    6    4

>> linspace(8,4,3)

ans =

   8    6    4
```

9. How many elements would be in the vectors created by thefollowing expressions?
    a. linspace(3,2000)

        100 (always, by default)

    b. logspace(3,2000)

        50 (always, by default – although these numbers would get very large quickly; most would be represented as Inf)

10. Create a variable myend which stores a random integer in the inclusive range from 5 to 9. Using the colon operator, create a vector that iterates from 1 to myend in steps of 3.

```
>> myend = randi([5, 9])

myend =

   9

>> vec = 1:3:myend

vec =

   1    4    7
```

11. Using the colon operator and the transpose operator, create a column vector myvec that has the values -1 to 1 in steps of 0.5.

```
>> rowVec = -1: 0.5: 1

rowVec =

  -1.0000   -0.5000         0    0.5000    1.0000

>> rowVec'

ans =
```

```
-1.0000
-0.5000
     0
 0.5000
 1.0000
```

12. Write an expression that refers to only the elements that have odd- numbered subscripts in a vector, regardless of the length of the vector. Test your expression on vectors that have both an odd and even number of elements.

```
>> vec = 1:8

vec =

   1    2    3    4    5    6    7    8

>> vec(1:2:end)

ans =

   1    3    5    7

>> vec = 4:12

vec =

   4    5    6    7    8    9   10   11   12

>> vec(1:2:end)

ans =

   4    6    8   10   12
```

13. Generate a 2 x 4 matrix variable mat. Replace the first row with 1:4. Replace the third column (you decide with which values).

```
>> mat = [2:5; 1 4 11 3]

mat =

   2    3    4    5
   1    4   11    3

>> mat(1,:) =     1:4

mat =

   1    2    3    4
   1    4   11    3

>> mat(:,3) =     [4;3]
```

```
mat =

   1    2    4    4
   1    4    3    3
```

14. Generate a 2 x 4 matrix variable *mat*. Verify that the number ofelements is the product of the number of rows and columns

```
>> mat = randi(20,2,4)

mat =

   6    5   14    4
  15    1    9    6

>> [r c] = size(mat)

r =

    2


c =

    4

>> numel(mat) == r * c

ans =

  logical

   1
```

15. Which would you normally use for a matrix: length or size? Why?

Size, because it tells you both the number of rows and columns.

16. When would you use length vs. size for a vector?

If you want to know the number of elements, you'd use length. If you want to figure out whether it's a row or column vector, you'd use size.

17. Generate a 2 x 3 matrix of random
    a. real numbers, each in the range (0, 1)

```
>> rand(2,3)

ans =

   0.0334    0.0782    0.7775
   0.0746    0.4518    0.1108
```

b. real numbers, each in the range (0, 10)

```
>> rand(2,3)*10

ans =

   0.0534   6.6795   6.0724
   9.5356   6.9165   9.2982
```

c. integers, each in the inclusive range from 5 to 20

```
>> randi([5, 20],2,3)

ans =

   17   16   13
   14   11   14
```

18. Create a variable rows that is a random integer in the inclusive range from 1 to 5. Create a variable cols that is a random integer in the inclusive range from 1 to 5. Create a matrix of all zeros with the dimensions given by the values of rows and cols.

```
>> rows = randi([1,5])

rows =

    3

>> cols = randi([1,5])

cols =

    2

>> zeros(rows,cols)

ans =

    0    0
    0    0
    0    0
```

19. Create a matrix variable mat. Find as many expressions as you can that would refer to the last element in the matrix, without assuming that you know how many elements or rows or columns it has (i.e., make your expressions general).

```
>>  mat = [12:15; 6:-1:3]

mat =

   12   13   14   15
    6    5    4    3

>> mat(end,end)
```

```
ans =

   3

>> mat(end)

ans =

   3

>> [r c] = size(mat)

r =

   2


c =

   4

>> mat(r,c)

ans =

   3
```

20. Create a vector variable vec. Find as many expressions as you can that would refer to the last element in the vector, without assuming that you know how many elements it has (i.e., make your expressions general).

```
>> vec = 1:2:9

vec =

   1    3    5    7    9

>> vec(end)

ans =

   9

>> vec(numel(vec))

ans =

   9

>> vec(length(vec))
```

```
ans =

    9

>> v = fliplr(vec)

v =

    9    7    5    3    1

>> v(1)

ans =

    9
```

21. Create a 2 x 3 matrix variable mat. Pass this matrix variable to each of the following functions and make sure you understand the result: flip, fliplr, flipud, and rot90. In how many different ways can you reshape it?

```
>> mat = randi([1,20], 2,3)

mat =

   19   20    3
    8   19   10

>> flip(mat)

ans =

    8   19   10
   19   20    3

>> fliplr(mat)

ans =

    3   20   19
   10   19    8

>> flipud(mat)

ans =

    8   19   10
   19   20    3

>> rot90(mat)

ans =
```

```
    3    10
   20    19
   19     8

>> rot90(rot90(mat))

ans =

   10    19     8
    3    20    19

>> reshape(mat,3,2)

ans =

   19    19
    8     3
   20    10

>> reshape(mat,1,6)

ans =

   19     8    20    19     3    10

>> reshape(mat,6,1)

ans =

   19
    8
   20
   19
    3
   10
```

22. What is the difference between fliplr(mat) and mat = fliplr(mat)?

fliplr(mat) stores the result in ans so mat is not changed.
mat = fliplr(mat)changes mat.

23. Use reshape to reshape the row vector 1:4 into a 2x2 matrix; store this in a variable named mat. Next, make 2x3 copies of mat using both repelem and repmat.

```
>> mat = reshape(1:4,2,2)

mat =

    1    3
    2    4

>> repelem(mat,2,3)
```

```
ans =

   1   1   1   3   3   3
   1   1   1   3   3   3
   2   2   2   4   4   4
   2   2   2   4   4   4

>> repmat(mat,2,3)

ans =

   1   3   1   3   1   3
   2   4   2   4   2   4
   1   3   1   3   1   3
   2   4   2   4   2   4
```