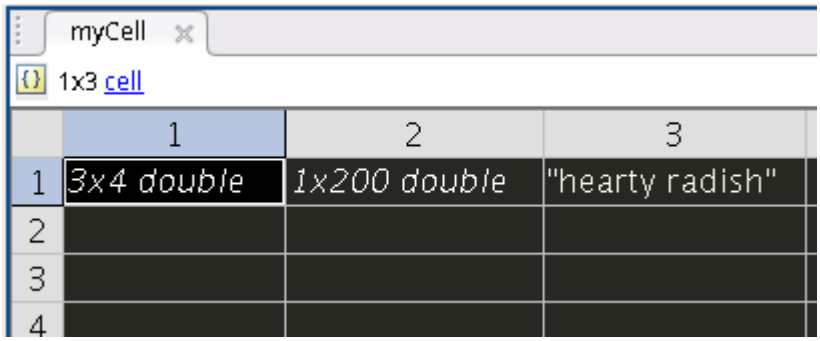
Practical Problems 3 – Data Structures, Formatting & Functions

Create a single script file for questions 1 - 4.

1. Create a cell array called ‘myCell’ where the first element is a random 3 x 4 matrix of integers, the second element is a vector with 200 elements ranging from -10 to 10, and the third element is a string of your choice.



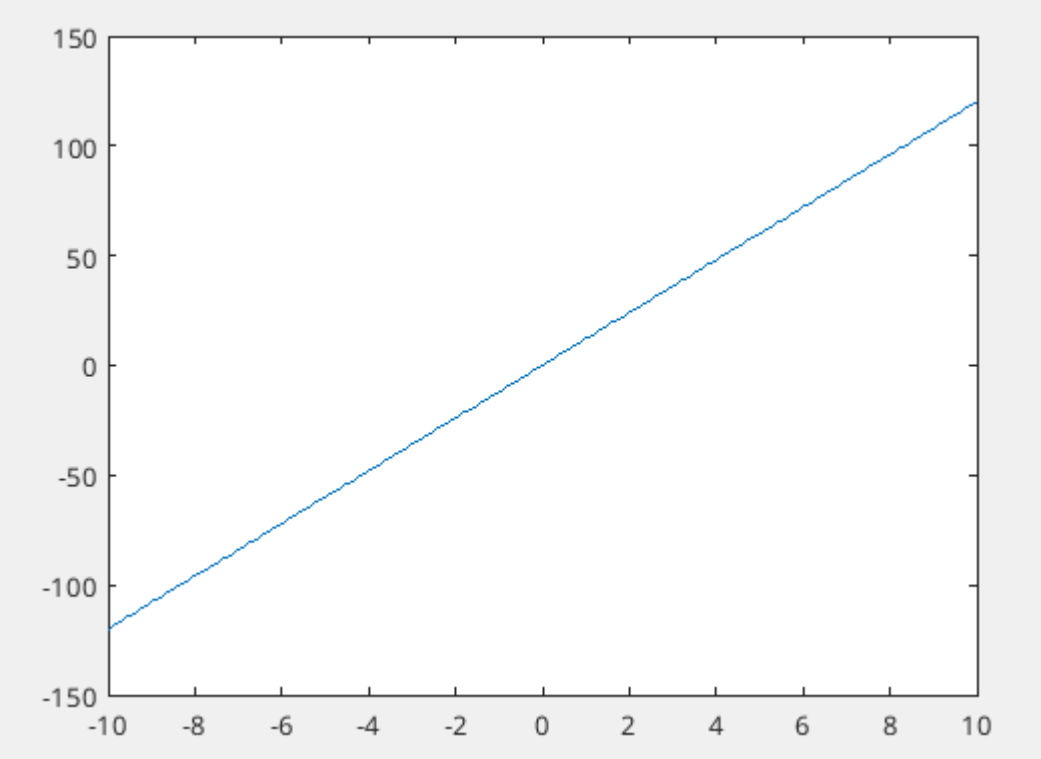
2. Request user input for indexes of an element in the 3 x 4 matrix. Your code should check that the user entered a valid pair of numbers and display a warning message if not.

> If the user enters 2 and 3 then the output should be the element in the matrix on the 2nd row 3rd column.

3. Create a new vector that is the vector in the cell array multiplied by the number in the matrix specified by the user, then insert this new vector as the third element of the cell array and shift the previous third element (string) to the fourth position of the cell array.

> For example if the user requests the matrix element that has value 12, then the new vector will be 12 times the original vector in the cell. Your cell array would then look like:
>
> myCell =
>
> 1×3 cell array
>
> {3×4 double} {1×200 double} {1×200 double} {["hearty radish"]}

4. Plot the 2 vectors from your cell array against each other in one line of code.



Create a single script file for questions 5 - 10.

5. Create a structure called myStruct to save the following data:

Name	**Age**	**Height (cm)**	**Mass (kg)**
Harry	36	170	80
Georgia	21	181	70
Elizabeth	78	158	65

myStruct

1x3 struct with 4 fields

Fields	Name	Age	Height_cm	Mass_kg
1	"Harry"	36	170	80
2	"Georg..."	21	181	70
3	"Elizab..."	78	158	65

6. Change Georgia's mass to be 68 kg.

myStruct

1x3 struct with 4 fields

Fields	Name	Age	Height_cm	Mass_kg
1	"Harry"	36	170	80
2	"Georg...	21	181	68
3	"Elizab...	78	158	65

7. Add an extra person to the data with the following details: Name, Lily; Age, 24; Height, 162 cm; Mass, 60 kg.

myStruct

1x4 struct with 4 fields

Fields	Name	Age	Height_cm	Mass_kg
1	"Harry"	36	170	80
2	"Georg...	21	181	68
3	"Elizab...	78	158	65
4	"Lily"	24	162	60

8. Using one line of code, calculate the mean height of the group.

mean_height

1x1 double

	1
1	169.6667

9. BMI is given by Mass (kg) divided by Height (metres) squared. Calculate the BMI of each person in the group and make a new field in your structure that saves this data for each person.

myStruct

1x4 struct with 5 fields

Fields	Name	Age	Height_cm	Mass_kg	BMI
1	"Harry"	36	170	80	27.6817
2	"Georg...	21	181	68	20.7564
3	"Elizab...	78	158	65	26.0375
4	"Lily"	24	162	60	22.8624

10. Use the sprintf function to display the names of the people in the group along with their age, height, mass and BMI displaying each value to 2 decimal places).

To format as shown below you must read about comma-separated lists using cell arrays in the documentation.

To align the names on the left and make the ages start at the same place use a negative value for the width of the string in the formatting.

```
Command Window
>> Q5_10
Name          Age        Height (cm)      Weight (kg)       BMI
Harry         36             170              80          27.68
Georgia       21             181              68          20.76
Elizabeth     78             158              65          26.04
Lily          24             162              60          22.86
```

11. Write a script that displays the number 12345.987654321 in the following formats (include leading 0's if the width is longer than the number):

(a) Width 10 precision 3.

(b) Width 10 precision 6.

(c) Width 8 precision 6.

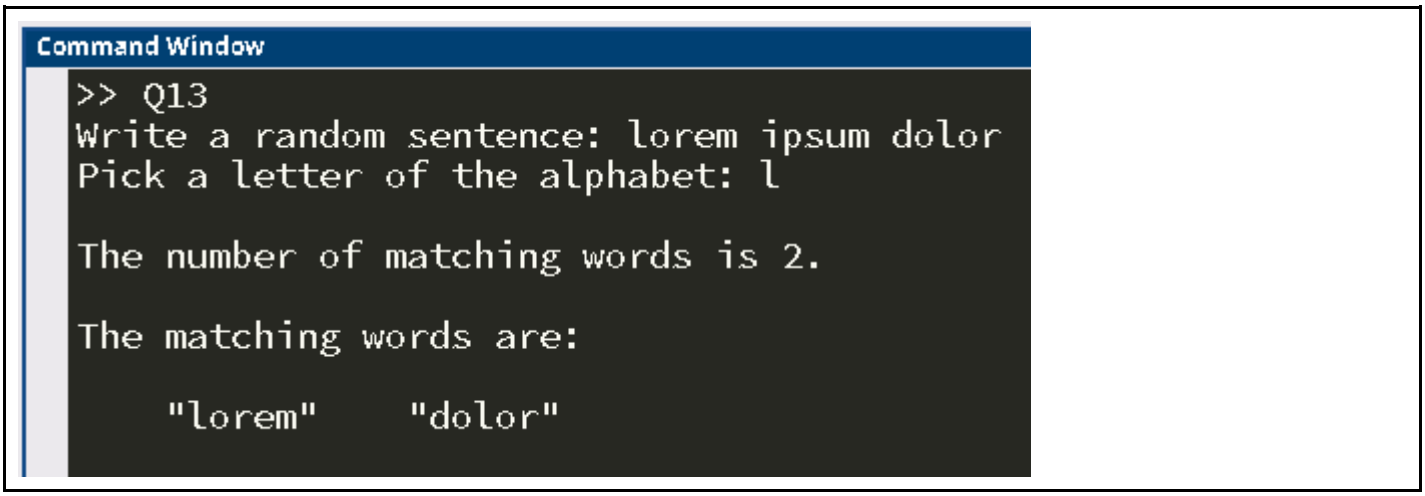
(d) Width 14 precision 6.

(e) What happens if you use a higher precision than there are decimal places (e.g. precision 12)?

12. Write a script that finds and displays the largest x that can be input to the exponential function (e^x) before an infinite result is reached.

```
Command Window
>> Q12
The largest input to the exponential function is: 709
```

13. Write a script that asks the user for 2 inputs. Prompt the user for a random sentence. Then prompt the user for a letter of the alphabet. Your script should then display how many times that letter appears in their sentence and lists the words in which it appears.



14. Load the datafile “DOB.mat” into the Matlab workspace then create a categorical array of the generations defined as follows:

“Boomer” born in 1946-1964
“Gen X” born in 1965-1980
“Millennials” born in 1981-1996
“Gen Z” born in 1997-1997-2012
“Gen A” born in 2012-present

(Hint: Use the discretize() function)

Display how many of each generation are in the data set then create a table that contains columns for each day, month, date, year and generation.
Display the first 10 table entries.
Now display the date of births of all Gen A.
Lastly, locate all people born in July and display the corresponding year of their births, the day on which they were born, and their generation.

Summary:

Boomer	137
Gen X	99
Millenials	102
Gen Z	96
Gen A	66

First 10 entries:

Day	Month	Date	Year	Gen
Thursday	July	7	1955	Boomer
Monday	October	22	1951	Boomer
Wednesday	February	14	2018	Gen A
Saturday	November	30	2002	Gen Z
Tuesday	June	21	1994	Millenials
Saturday	August	23	1997	Gen Z
Friday	January	10	1958	Boomer
Friday	April	3	1953	Boomer
Monday	November	11	1946	Boomer
Friday	September	13	2013	Gen A

Gen A:

Day	Month	Date	Year	Gen
Wednesday	February	14	2018	Gen A
Friday	September	13	2013	Gen A
Monday	January	9	2012	Gen A
Thursday	June	11	2020	Gen A
Friday	July	17	2020	Gen A
Saturday	June	15	2013	Gen A
Sunday	February	10	2019	Gen A
Sunday	April	26	2015	Gen A
Thursday	July	31	2014	Gen A
Thursday	May	30	2013	Gen A
Tuesday	December	18	2018	Gen A
Saturday	March	7	2020	Gen A
Saturday	September	5	2015	Gen A
Wednesday	June	5	2013	Gen A
Monday	October	8	2012	Gen A
Wednesday	August	15	2018	Gen A
Monday	April	21	2014	Gen A
Saturday	December	17	2016	Gen A

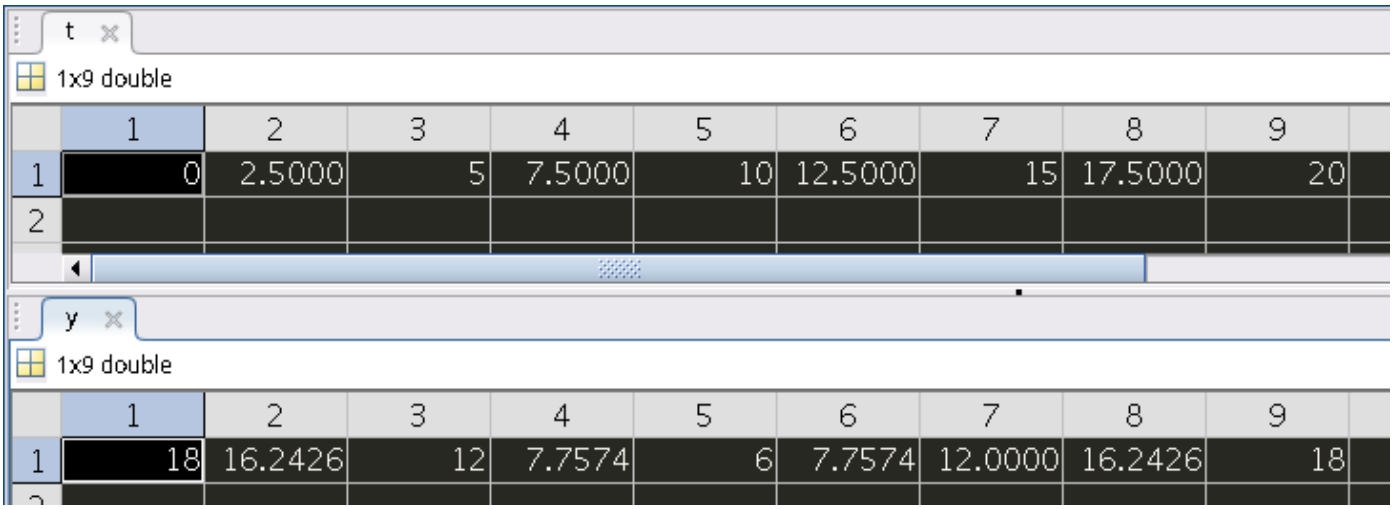
Sunday	May	3	2015	Gen A
Monday	December	12	2016	Gen A
Monday	April	3	2017	Gen A
Friday	October	26	2012	Gen A
Tuesday	May	26	2015	Gen A
Friday	October	19	2012	Gen A
Tuesday	March	12	2013	Gen A
Monday	August	8	2016	Gen A
Monday	February	26	2018	Gen A
Thursday	February	20	2014	Gen A
Tuesday	October	28	2014	Gen A
Sunday	January	10	2021	Gen A
Sunday	July	24	2016	Gen A
Tuesday	April	17	2018	Gen A
Wednesday	September	12	2012	Gen A
Friday	November	21	2014	Gen A
Thursday	September	12	2019	Gen A
Wednesday	August	29	2012	Gen A
Friday	May	15	2015	Gen A
Thursday	September	11	2014	Gen A
Sunday	June	3	2012	Gen A
Sunday	June	12	2016	Gen A
Monday	September	30	2013	Gen A
Tuesday	October	3	2017	Gen A
Thursday	September	13	2018	Gen A
Thursday	June	21	2012	Gen A
Sunday	April	30	2017	Gen A
Sunday	December	21	2014	Gen A
Saturday	March	10	2012	Gen A
Thursday	July	19	2012	Gen A
Sunday	August	5	2018	Gen A
Monday	January	16	2017	Gen A
Friday	March	29	2013	Gen A
Saturday	March	1	2014	Gen A
Friday	January	25	2019	Gen A
Wednesday	May	27	2015	Gen A
Tuesday	May	23	2017	Gen A
Thursday	May	31	2012	Gen A
Saturday	April	5	2014	Gen A
Saturday	March	10	2018	Gen A
Sunday	May	4	2014	Gen A
Sunday	June	10	2012	Gen A
Monday	October	12	2020	Gen A
Monday	March	5	2012	Gen A

```
    Monday        March         5       2012    Gen A
    Monday        August        24      2020    Gen A
    Wednesday     October       21      2020    Gen A
    Monday        October       26      2020    Gen A
    Wednesday     January       16      2019    Gen A


People born in July:
--------------------
    Year        Day          Gen
    ----    ---------    ----------

    1955    Thursday     Boomer
    2020    Friday       Gen A
    1964    Saturday     Boomer
    2014    Thursday     Gen A
    1979    Friday       Gen X
    1957    Tuesday      Boomer
    1977    Monday       Gen X
    1947    Wednesday    Boomer
    2004    Wednesday    Gen Z
    1983    Thursday     Millenials
    1975    Saturday     Gen X
    2006    Wednesday    Gen Z
    1953    Friday       Boomer
    2016    Sunday       Gen A
    1957    Tuesday      Boomer
    1984    Sunday       Millenials
    1970    Sunday       Gen X
    1961    Thursday     Boomer
    1950    Saturday     Boomer
    1957    Friday       Boomer
    2005    Sunday       Gen Z
    1946    Tuesday      Boomer
    1948    Tuesday      Boomer
    1953    Thursday     Boomer
    1962    Thursday     Boomer
    2012    Thursday     Gen A
    1992    Friday       Millenials
    1956    Thursday     Boomer
    1957    Friday       Boomer
    1968    Saturday     Gen X
    1966    Wednesday    Gen X
    1992    Wednesday    Millenials
    1971    Monday       Gen X

>>
```

15. The following code performs some tasks in Matlab using a for loop. Vectorise the code to produce the same result without using any loops.

```
tstart=0; tend=20; ni=8;
t(1)=tstart;
y(1)=12 + 6*cos(2*pi*t(1)/(tend-tstart));
for i=2:ni+1
  t(i)=t(i-1)+(tend-tstart)/ni;
  y(i)=12 + 6*cos(2*pi*t(i)/ ...
       (tend-tstart));
end
```

The for loop creates the vectors shown below. Create both vectors using a single line of code for each.



t

1x9 double

	1	2	3	4	5	6	7	8	9
1	0	2.5000	5	7.5000	10	12.5000	15	17.5000	20
2									

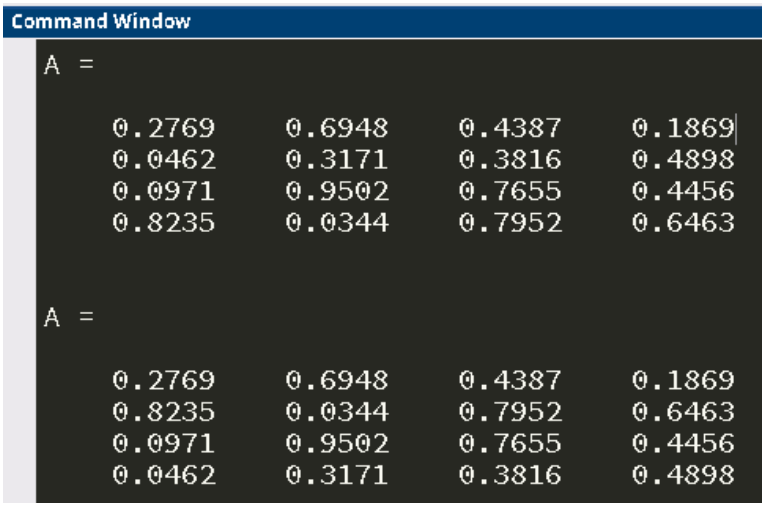
y

1x9 double

	1	2	3	4	5	6	7	8	9
1	18	16.2426	12	7.7574	6	7.7574	12.0000	16.2426	18

16. Create a random 4 x 4 matrix, ***A***, with values between 0 and 1 then swap the 2nd and 4th row using one line of code.

(example solution)

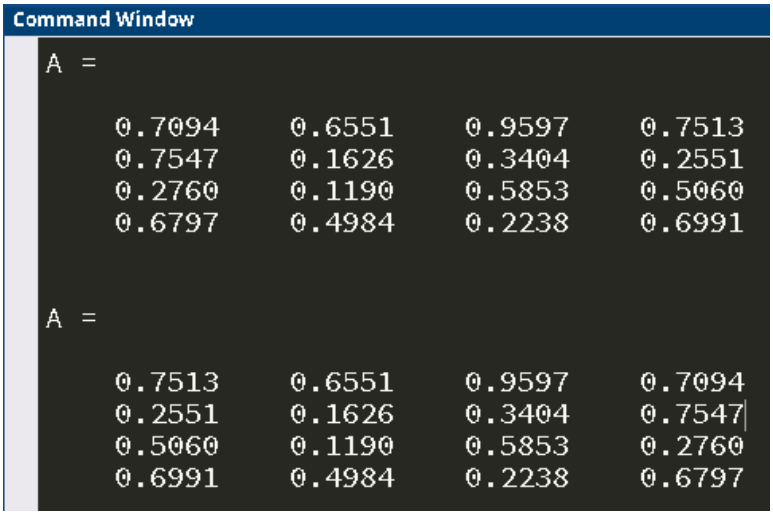


Command Window

```
A =

    0.2769    0.6948    0.4387    0.1869
    0.0462    0.3171    0.3816    0.4898
    0.0971    0.9502    0.7655    0.4456
    0.8235    0.0344    0.7952    0.6463


A =

    0.2769    0.6948    0.4387    0.1869
    0.8235    0.0344    0.7952    0.6463
    0.0971    0.9502    0.7655    0.4456
    0.0462    0.3171    0.3816    0.4898
```

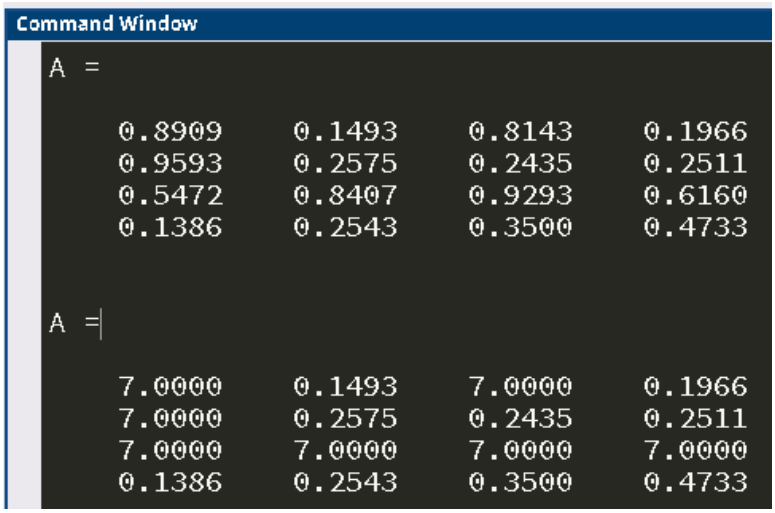
17. Using the matrix, ***A***, from Q16 now swap the 1st and 4th columns using one line of code.

(example solution)



18. Using one line of code, set every value of ***A*** that is greater than 0.5 equal to 7.

(example solution)



19. Create a 4 x 7 matrix, ***B***, of random integers between 10 and 50 then using one line of code set every value of ***B*** that is both greater than 20 and less than 40 equal to 0 (***Hint:*** Use the element-wise logical operators, '&' and '|').

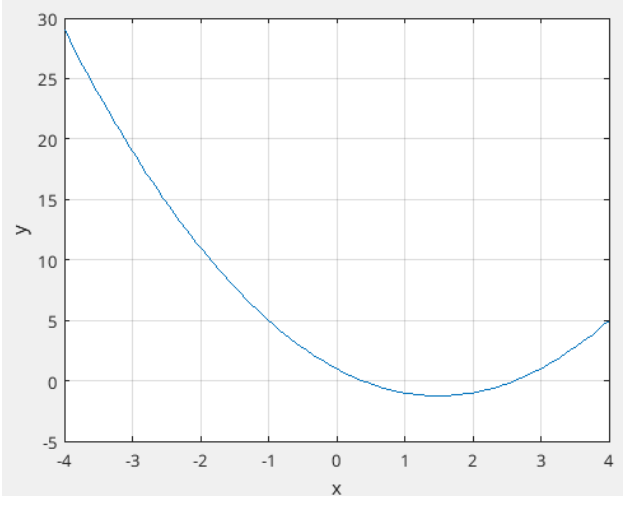
(example solution)

```
Command Window
B =

    24    47    25    31    33    16    16
    44    21    33    41    29    42    34
    33    41    13    48    10    22    20
    32    40    12    15    23    31    36


B =

     0    47     0     0     0    16    16
    44     0     0    41     0    42     0
     0    41    13    48    10     0    20
     0    40    12    15     0     0     0
```

20. Using one line of code insert the column vector $[3\ -2\ 4\ 8]^T$ in between the 2nd and 3rd columns of ***B***.

```
Command Window
B =

    45    44    26    17    47    23    41
    33    35    13    19    48    46    25
    32    24    19    27    30    25    19
    15    31    15    12    30    14    26


B =

    45    44     0    17    47     0    41
     0     0    13    19    48    46     0
     0     0    19     0     0     0    19
    15     0    15    12     0    14     0


B =

    45    44     3     0    17    47     0    41
     0     0    -2    13    19    48    46     0
     0     0     4    19     0     0     0    19
    15     0     8    15    12     0    14     0
```

21. Write a script that creates an anonymous function for the function $f(x) = x^2 - 3x + 1$ then plots it between -4 and 4 with a grid.



22. Write a script that creates 2 anonymous functions, f1 and f2, for sin(x) and cos(x), then a 3rd anonymous function, f3, equal to the first function divided by the second function (f1/f2 = sin(x)/cos(x) = tan(x)). The script should then plot all 3 functions in different colours using subplots over the domain $-2\pi < x < 2\pi$. Set the axes on the graph to have y-limits between -1 and 1.

For this question make sure you do not define:

f3 = @(x) tan(x)

You must use f1 and f2 in your f3 function.

23. Create an anonymous function, f4, that takes 2 inputs, x and y, then calculates x^y. Test your function with inputs x = 3 and y = 7.

ans =

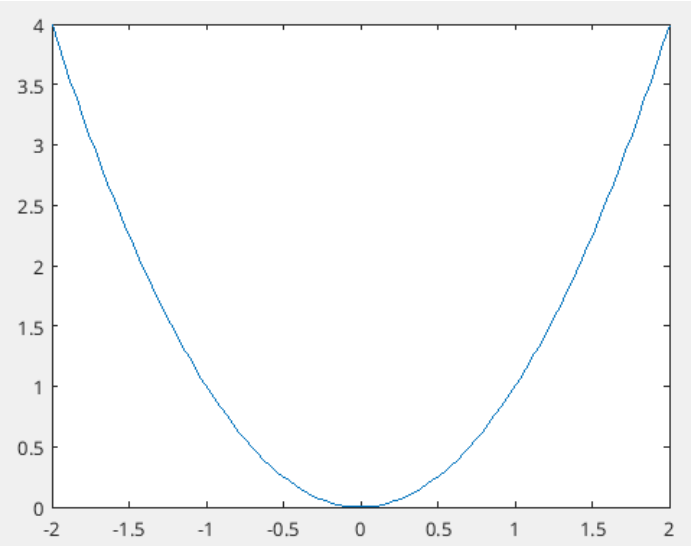
2187

24. Create an anonymous function, f5, that takes 3 inputs, f, a and b, where f is another anonymous function, and plots f between a and b. Test f5 by using any test function over whatever domain you like.

(example solution)

I chose to test my function by doing:

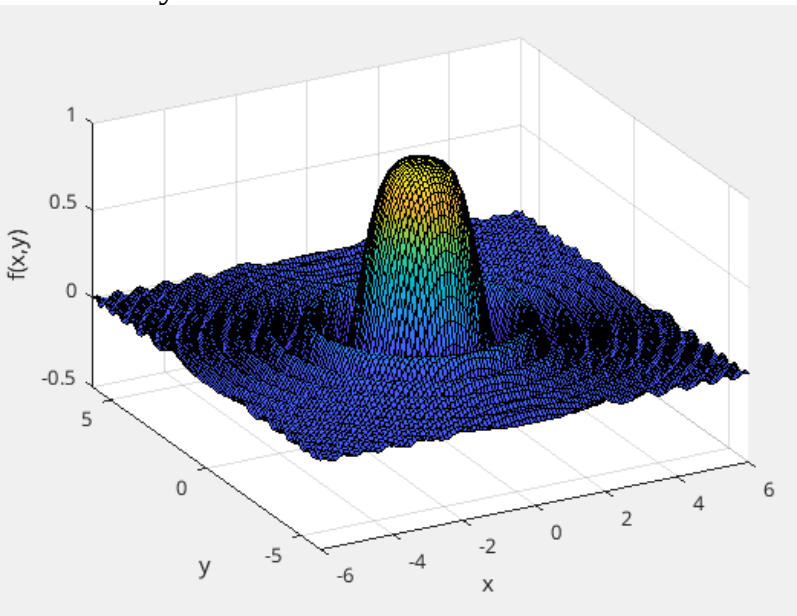
>> f5(@(x) x.^2, -2, 2)



25. Create a function file (**not anonymous**) that takes 3 inputs and produces a surface plot. The first input should be a function that accepts 2 arguments (x and y), the second and third inputs should be vectors containing the desired values of x and y over which to plot. Test your function using any function and domain.

(example solution)

I tested my function file with an input function $\sin(x^2+y^2)/(x^2+y^2)$ between -6 and 6 on both the x- and y-axes.



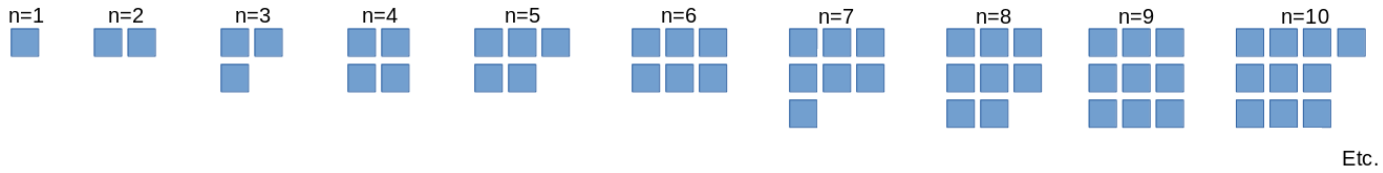
26. Write a function file that takes between 3 and 6 input arguments (all scalar values) and returns a vector containing all the inputs sorted from lowest to highest. Warning messages should be displayed in the command window if the user enters the wrong number of inputs, or a vector/string instead of a scalar.

```
Command Window
>> Q26(1,2,3)
ans =
     6
>> Q26(1,2,3,4)
ans =
    10
>> Q26(1,2,3,4,5)
ans =
    15
>> Q26(1,2,3,4,5,6)
ans =
    21
>> Q26(1,2,3,4,5,6,7)
Warning: Too many inputs. Summing the first 6.
> In Q26 (line 6)
ans =
    21
>> Q26(1,2,3,'sdf')
Error using Q26 (line 10)
Input must be a number.
>> Q26(1,2,'sd',4)
Error using Q26 (line 10)
Input must be a number.
```

27. ## Challenge Problem

Write a function file that requires 3 inputs, *F*, *a* and *b*, where *F* is a function of one variable, and plots the function between *a* and *b*, but also accepts additional optional input arguments that specify other intervals to plot the function over.

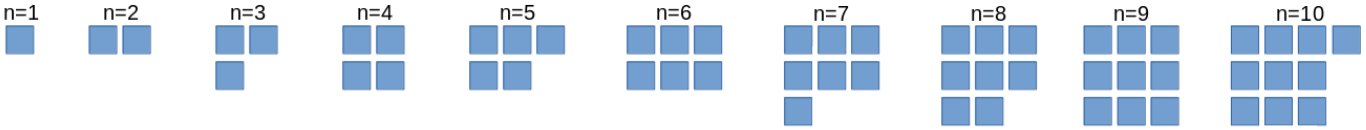
The plots should appear in the same figure (subplots) with the pattern as shown below:



Note that the additional optional arguments should come in pairs (*c* and *d*, *e* and *f etc.*). Display warning messages if the user enters the wrong number or type of input. Finally accept up to 100 subplots.
Hint: To determine if the first argument is a function read about **isa()** in the documentation.
Hint: To determine the layout of the subplots it will be helpful to think about the relationship between square numbers, number of plots requested (n), and the number of rows and columns.

Write a function file that requires 3 inputs, *F*, *a* and *b*, where *F* is a function of one variable, and plots the function between *a* and *b*, but also accepts additional optional input arguments that specify other intervals to plot the function over. The plots should appear in the same figure (subplots). Note that the additional optional arguments should come in pairs (*c* and *d*, *e* and *f etc.*). Display warning messages if the user enters the wrong number or type of input. Finally accept up to 100 subplots.
Hint: To determine if the first argument is a function read about isa() in the documentation.
Hint: To make the layout of the plots dynamic (responds to how many subplots you request),

try organising them into the following pattern:

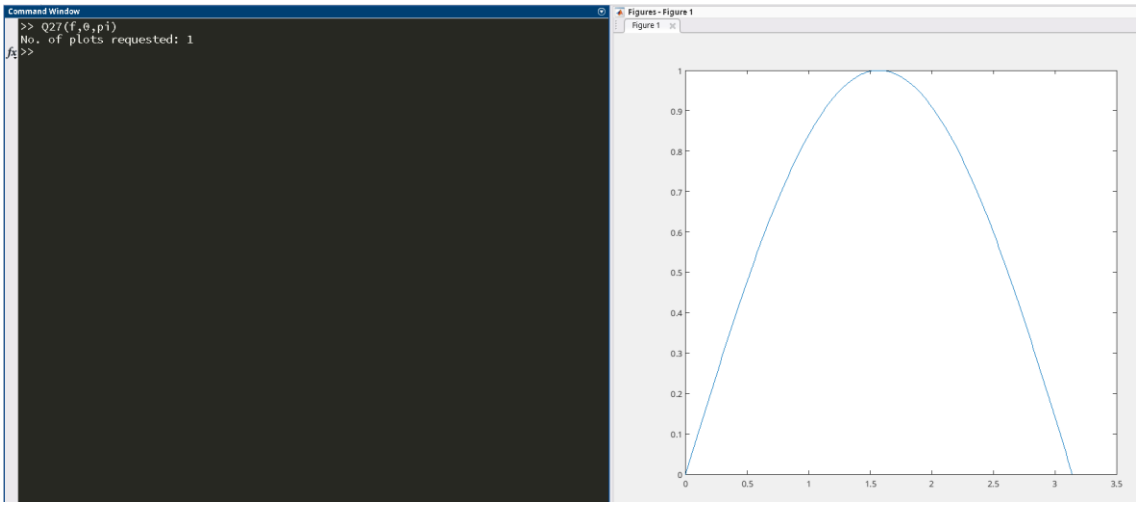


Etc.

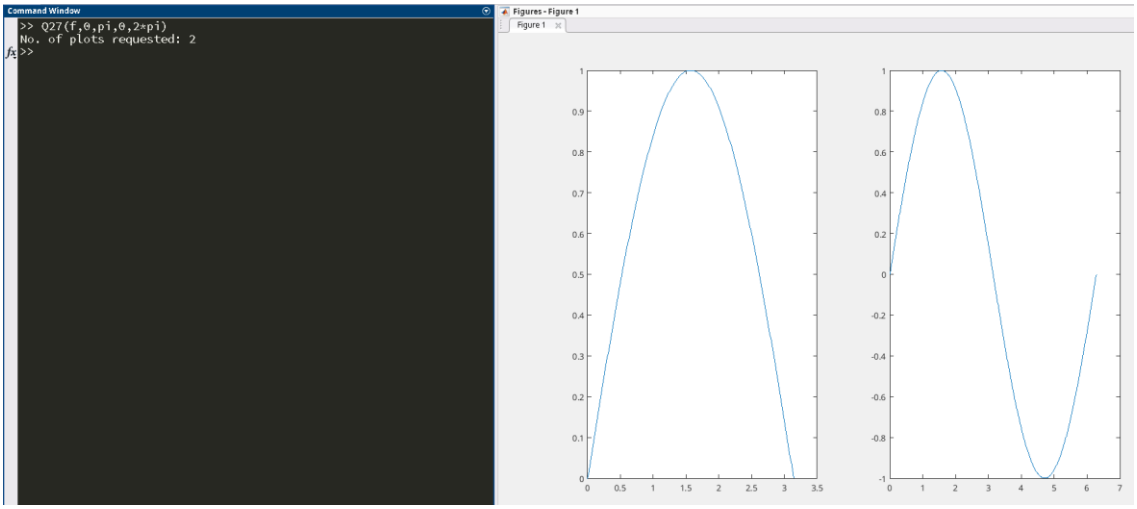
Test this function with sin(x) over varying intervals. Try to test if your code catches input errors too.

Testing different numbers of subplots follow the pattern:

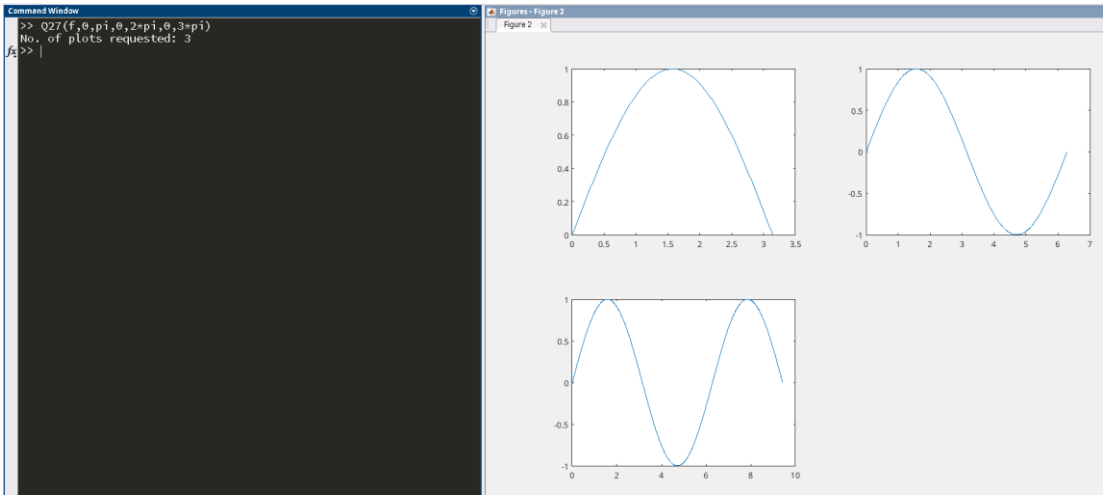
- 1 interval



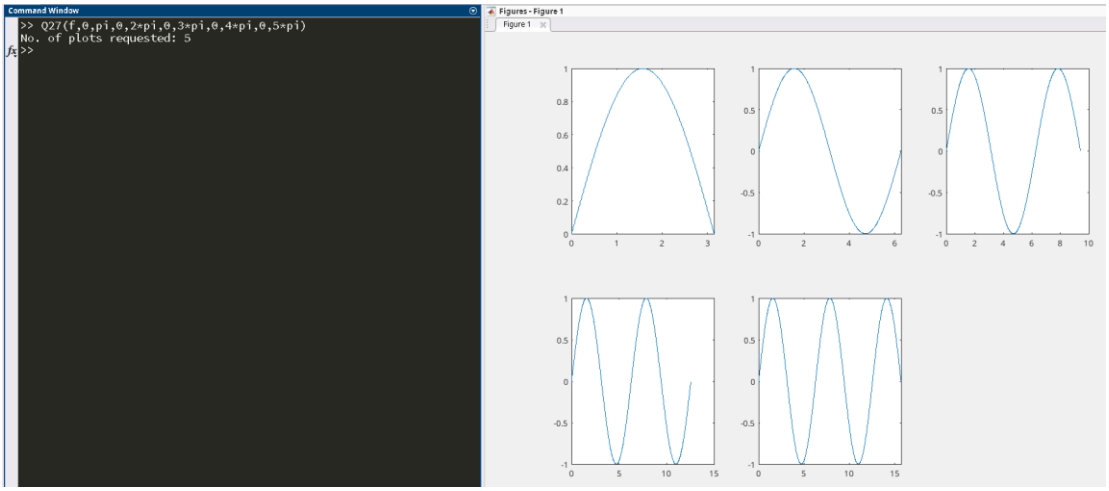
- 2 intervals



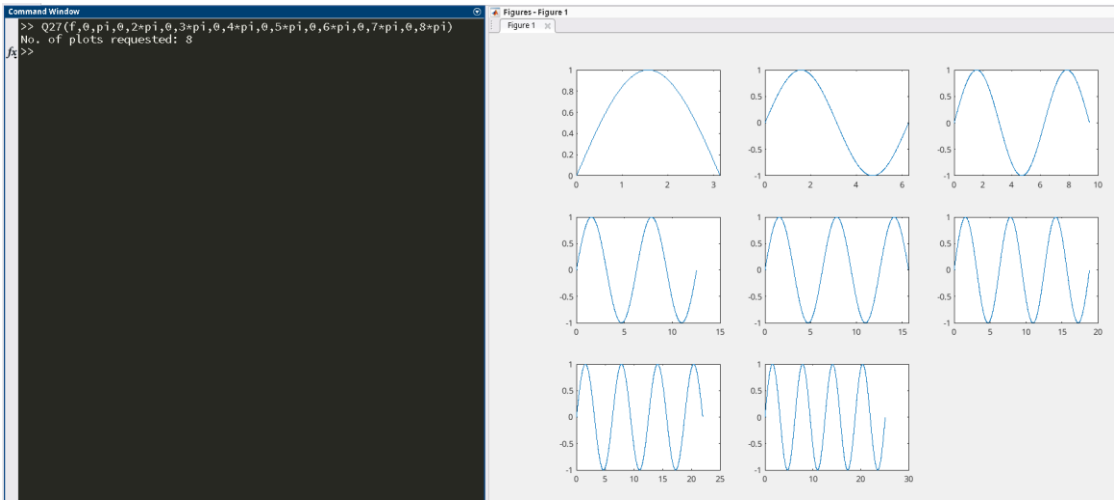
- 3 intervals



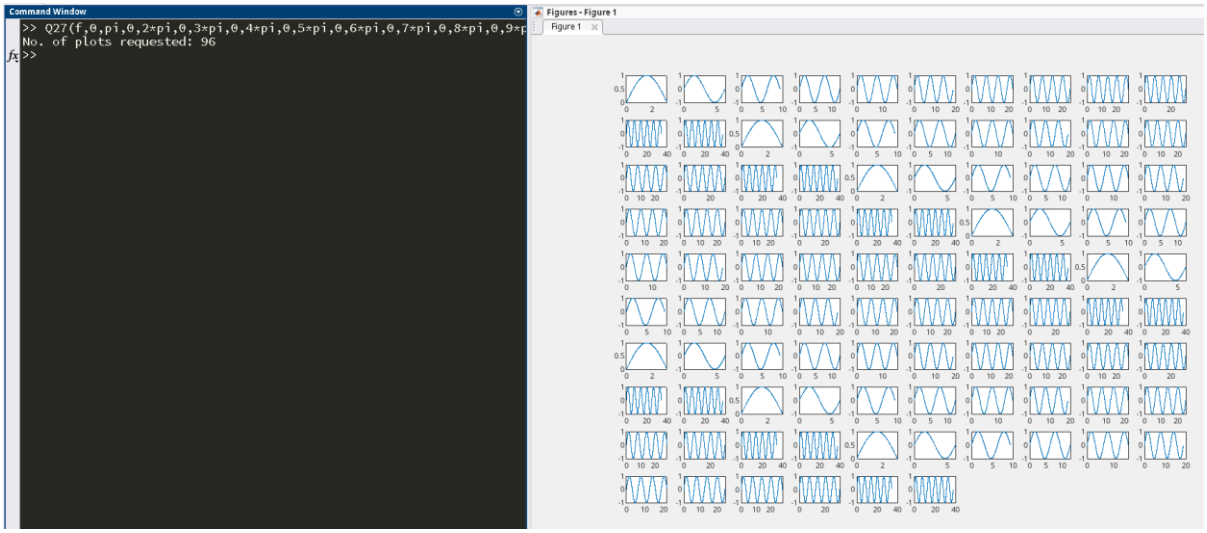
- 5 intervals



- 8 intervals



- 96 intervals



Testing that input errors are handled:

- Too many inputs

```
Command Window
>> Q27(f,0,pi,0,2*pi,0,3*pi,0,4*pi,0,5*pi,0,6*pi,0,7*pi,0,8*pi,0,9*pi,0,10*pi,0,11*pi
No. of plots requested: 101
Error using Q27 (line 16)
Either odd number of x-value inputs or there are too many. Maximum is 100 subplots.
```

- Odd number of interval inputs

```
Command Window
>> Q27(f,0,pi,0)
No. of plots requested: 1.5
Error using Q27 (line 16)
Either odd number of x-value inputs or there are too many. Maximum is 100 subplots.
```

- First argument not a function

```
Command Window
>> Q27('NotaFunction',0,pi)
Error using Q27 (line 19)
Please input a function for the first argument followed by numbers for the x-value pairs.
```

- An interval argument is not a number

```
Command Window
>> Q27(f,0,'NotaNumber')
Error using Q27 (line 19)
Please input a function for the first argument followed by numbers for the x-value pairs.
```