Write a script that will generate a vector of 10 random integers, each in the inclusive range from 0 to 100. If the integers are evenly distributed in this range, then when arranged in order from lowest to highest, they should fall on a straight line. To test this, fit a straight line through the points and plot both the points and the line with a legend.

Proj.1 T. Mallahi, Z. Aminahesari, S. Arabi, S. Khandouzi

Write a function that will receive data points in the form of x and y vectors. If the lengths of the vectors are not the same, then they can't represent data points so an error message should be printed. Otherwise, the function will fit a polynomial of a random degree through the points, and will plot the points and the resulting curve with a title specifying the degree of the polynomial. The degree of the polynomial must be less than the number of data points, n, so the function must generate a random integer in the range from 1 to n-1 for the polynomial degree.

Write a function that will receive a variable number of input arguments: the length and width of a rectangle, and possibly also the height of a box that has this rectangle as its base. The function should return the rectangle area if just the length and width are passed, or also the volume if the height is also passed.

Write a function called “makemat” that will receive two row vectors as input arguments, and from them create and return a matrix with two rows. You may not assume that the length of the vectors is known. Also, the vectors may be of different lengths. If that is the case, add 0’s to the end of one vector first to make it as long as the other. For example, a call to the function might be:

```
>>makemat(1:4, 2:7)
ans =

     1     2     3     4     0     0
     2     3     4     5     6     7
```

Proj. 2 A. Erfani, T. Kalaee, A. Banihashemi, M. Hesari

The inverse of the mathematical constant *e* can be approximated as follows:

$$\frac{1}{e} \approx \left(1 - \frac{1}{n}\right)^n$$

Write a script that will loop through values of *n* until the difference between the approximation and the actual value is less than 0.0001. The script should then print out the built-in value of e^{-1} and the approximation to 4 decimal places, and also print the value of *n* required for such accuracy.

Write a function *plot2fnhand* that will receive two function handles as input arguments, and will display in two Figure Windows plots of these functions, with the function names in the titles. The function will create an x vector that ranges from 1 to n (where n is a random integer in the inclusive range from 4 to 10). For example, if the function is called as follows

```
>> plot2fnhand(@sqrt, @exp)
```

and the random integer is 5, the first Figure Window would display the **sqrt** function of x = 1:5, and the second Figure Window would display **exp(x)** for x = 1:5.

The Wind Chill Factor (WCF) measures how cold it feels with a given air temperature T (in degrees Fahrenheit) and wind speed V (in miles per hour). One formula for WCF is

$$WCF = 35.7 + 0.6\,T - 35.7\,(V^{0.16}) + 0.43\,T\,(V^{0.16})$$

Write a function to receive the temperature and wind speed as input arguments, and return the WCF. Using loops, print a table showing wind chill factors for temperatures ranging from -20 to 55 in steps of 5, and wind speeds ranging from 0 to 55 in steps of 5. Call the function to calculate each wind chill factor.

Instead of printing the WCFs in the previous problem, create a matrix of WCFs and write them to a file. Use the programming method, using nested loops.

Proj. 3. M. Amouee, S. Ghasemi, M.H. Dankoub, A. Jafaee

The number of faculty members in each department at a certain College of Engineering is:

```
ME  22
BM  45
CE  23
EE  33
```

Experiment with at least 3 different plot types to graphically depict this information. Make sure that you have appropriate titles, labels, and legends on your plots. Which type(s) work best, and why?

The lump sum S to be paid when interest on a loan is compounded annually is given by $S = P(1 + i)^n$ where *P* is the principal invested, i is the interest rate, and *n* is the number of years. Write a program that will plot the amount S as it increases through the years from 1 to *n*. The main script will call a function to prompt the user for the number of years (and error-check to make sure that the user enters a positive integer). The script will then call a function that will plot S for years 1 through n. It will use .05 for the interest rate and $10,000 for P.

Proj. 4, J. Sepehnia, H. Mirsaraee, A. Shekarian, M. Mohammadbagheri

Write a function that will receive x and y vectors representing data points. The function will create, in one Figure Window, a plot showing these data points as circles and also in the top part a second-order polynomial that best fits these points and on the bottom a third-order polynomial. The top plot will have a line width of 3 and will be a gray color. The bottom plot will be blue, and have a line width of 2. For example, the Figure Window might look like this.

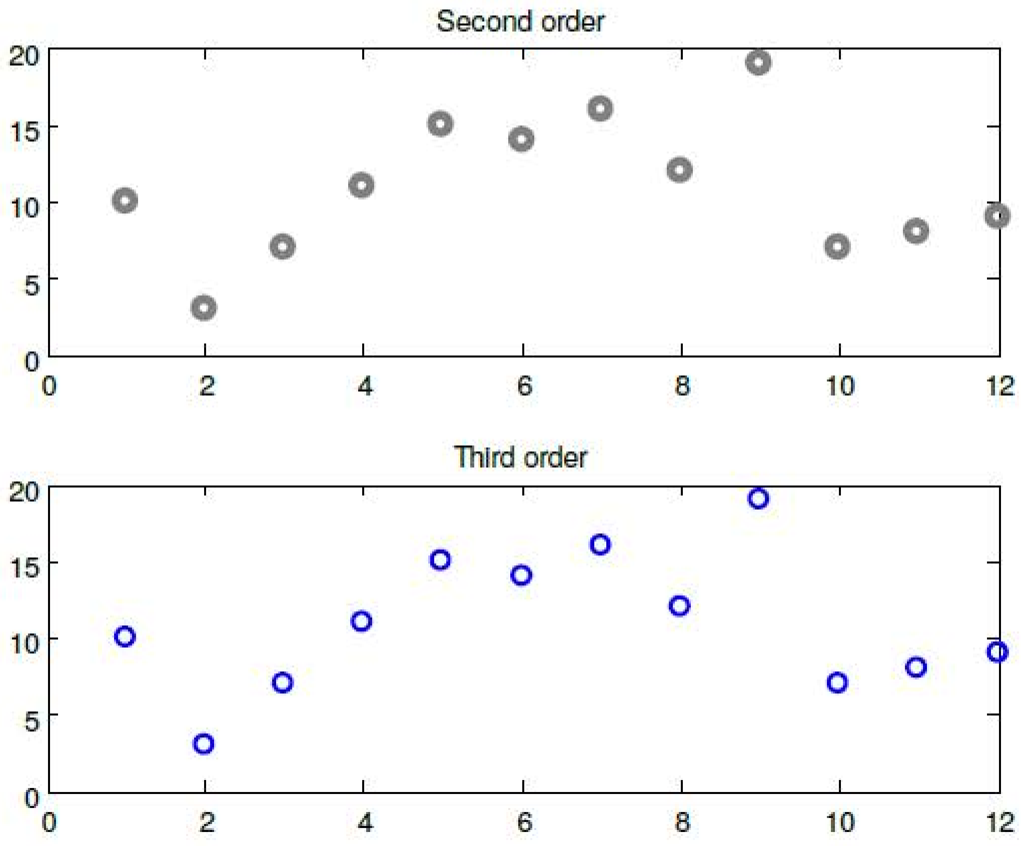

Figure Subplot of second and third order polynomials with different line properties

The axes are the defaults. Note that changing the line width also changes the size of the circles for the data points. You do not need to use a loop.

Write a menu-driven program to convert a time in seconds to other units (minutes, hours, and so on). The main script will loop to continue until the user chooses to exit. Each time in the loop, the script will generate a random time in seconds, call a function to present a menu of options, and print the converted time. The conversions must be made by individual functions (e.g. one to convert from seconds to minutes). All user-entries must be error-checked.

The Mystical River's water flow rate on a particular day is shown in the table below. The time is measured in hours and the water flow rate is measured in cubic feet per second. Write a script that will fit polynomials of degree 3 and 4 to the data and create a subplot for the two polynomials. Plot also the original data as black circles in both plots. The titles for the subplots should include the degree of the fitted polynomial. Also, include appropriate x and y labels for the plots.

Time	0	3	6	9	12	15	18	21	24
Flow Rate	800	980	1090	1520	1920	1670	1440	1380	1300