<u>Homework- chapter 1&2 (50 points)</u>

Text: Lent Chapter 1, Each problem is worth 10 points:

Problems: 1.1, 1.2, 1.5, 1.10, 1.18, 2.4, 2.11, 2.12, 2. 20, 2. 21

For each problem, Write a script/ program using the block cell format including:

Information header,

Display the problem #,

Set parameters,

Do Calculation,

Display results.

Choose appropriate variable names, Include useful comments and use spacing for increased readability.

Submission:

Once you have all your problems completed, copy them in order to a master script. Have a comment at the beginning with your name and chapter 1&2 HW. Have clc, clear be the next line. This will clear all the commands from the window and variables from the workspace. Before submitting your master script file named as "your name_chapter1", make sure that the programs are working. The problem number and results should be clearly displayed.

- Use comments appropriately so that a reader sees clearly what the program is doing. Specify the units of physical quantities.
- Use display statements (`disp`) to show both the problem input parameters and solution results. Pay attention to spaces and blank lines in formatting the output statements clearly.

Forming good programming habits pays off. Clear well-written code is easier to understand and change.

1. **Quadratic roots.** Write a program, `quadroots.m`, for finding the roots of the second-order polynomial $ax^2 + bx + c$. Use the quadratic equation. The inputs are the coefficients a, b, and c and the outputs are z_1 and z_2. The program should produce (exactly) the following output in the Command window when run with $(a, b, c) = (1, 2, -3)$:

```
====================
Quadratic Solver

coefficients
  a = 1
  b = 2
  c = -3
roots
  z1 = 1
  z2 = -3
```

2. **Rolling dice.** Write a program, `ThrowDice.m`, which generates the sum from randomly throwing a pair of fair dice. Use the `randi` function.
3. **Right triangle.** Write a program, `triangle.m`, that finds the length of the hypotenuse and the acute angles in a right triangle, given the length of the two legs of the right triangle.
4. **Running to the Sun.** Write a program, `sunrun.m`, to calculate how many hours it would take to run to the Sun if averaging a five-minute mile. Display the answer in seconds, hours, and years. (The distance from the Earth to the Sun is 93 million miles.)
5. **Gravitational force.** Write a program, `gforce.m`, to find the gravitational force (in Newtons) between any two people using $F = Gm_1m_2/r^2$ with gravitational constant $G = 6.67300 \times 10^{-11} N \cdot m^2/kg^2$. Run the program to find the gravitational force exerted on one person whose mass is 80 kg by another person of mass 60 kg who is 2 m away and report this value in the Command window.
6. **Shoes on the Moon.** Write a program, `moonshoes.m`, to estimate the number of shoes that it would take to cover the surface of the moon (which has a radius of 1740 km). Choose a shoe length and width.
7. **Kite string.** Write a program, `kitestring.m`, to calculate the amount of string needed to fly a kite at a height `kiteHeight` and at an angle `theta` to the horizon. Assume the person holds the kite a distance `holdHeight` above the ground and wants to have a minimum length of `stringWound` wound around the string holder. Run the program for a height of 8.2 m, an angle of $2\pi/7$, with 0.25 m of string around the holder, which is held 0.8 m above the ground.

8. **Calculating an average.** Write a program, `randavg.m`, that calculates the average of five random numbers between 0 and 10, which are generated using `rand`. Reset the random number generator using the `rng('shuffle')` command before finding the random numbers.

9. **Stellar parallax.** In the 16th century, Tycho Brahe argued against the Copernican heliocentric model of the universe (actually the solar system), because he reasoned that if the Earth moved around the Sun, you would see the apparent angular positions of stars shifting back and forth. This phenomenon is known as the stellar parallax, and it wasn't measured until the 19th century because it's so small. The stars are much farther away than anyone in the 16th century imagined. The star closest to the Sun is Proxima Centauri, which has an annual parallax $\delta\theta = 0.7$ arcseconds. That means that over six months the apparent position in the sky shifts by a maximum angle of seven-tenths of one 3600th of a degree. Write a program, parallax.m, to calculate how far away a one-inch diameter disk would need to be to subtend the same angle. Express the answer in feet and miles.

10. **Elastic collisions in one dimension.** When two objects collide in such a way that the sum of their kinetic energies before the collision is the same as the sum of their kinetic energies after the collision, they are said to collide elastically. The final velocities of the two objects can be obtained by using the following equations.

$$v_{1f} = \left(\frac{m_1 - m_2}{m_1 + m_2}\right) v_{1i} + \left(\frac{2m_2}{m_1 + m_2}\right) v_{2i}$$

$$v_{2f} = \left(\frac{2m_1}{m_1 + m_2}\right) v_{1i} + \left(\frac{m_2 - m_1}{m_1 + m_2}\right) v_{2i}$$

Write a program, `Collide.m`, that calculates the final velocities of two objects in an elastic collision, given their masses and initial velocities. Use $m_1 = 5$ kg, $m_2 = 3$ kg, $v_{1i} = 2$ m/s, and $v_{2i} = -4$ m/s as a test case.

11. **Kinetic friction.** The magnitude of the kinetic friction, F_{fric}, on a moving object is calculated using the equation:

$$F_{fric} = \mu_k N$$

where μ_k is the coefficient of kinetic friction, and N is the normal force on the moving object. If the object is on a surface parallel to the ground, the normal force is simply the weight of the object, $N = mg$, where $g = 9.8\ m/s^2$, the acceleration due to gravity, and mass is measured in kg. Write a program, `Friction.m`, which calculates the force of kinetic friction on a horizontally moving object, given its mass and the coefficient of kinetic friction. Run the program with each of the following sets of parameters:

a. $m = 0.8$ kg $\quad \mu_k = 0.68$ (copper and glass)

b. $m = 50$ g $\quad \mu_k = 0.80$ (steel and steel)

c. $m = 324$ g $\quad \mu_k = 0.04$ (Teflon and steel)

12. **Light travel time.** Write a program, `LightTime.m`, to calculate the time it takes light to travel (a) from New York to San Francisco, (b) from the Sun to the Earth, (c) from Earth to Mars (minimum and maximum), (d) from the Sun to Pluto.

13. Finite difference. Consider the function $f(x) = cos(x)$. Write a program, `FiniteDiff.m`, to calculate the finite-difference approximation to the derivative of $f(x)$ at the point $x = x_0$ from the expression:

$$\left.\frac{df(x)}{dx}\right]_{x=x_0} \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Run the program with smaller and smaller numbers for Δx to see that the approximation converges to a limit. Find the limit for various values of x_0, including $x_0 = 0, \frac{\pi}{4}, \frac{\pi}{2}, \pi, \frac{5\pi}{4}$, and 2π. Could you have predicted the results you get?

14. Total payment. The monthly payment, P, computed for a loan amount, L, that is borrowed for a number of months, n, at a monthly interest rate of c is given by the equation:

$$P = \frac{L * c(1+c)^n}{(1+c)^n - 1}$$

Write a program, `TotalPayment.m`, to calculate the total amount of money a person will pay in principal and interest on a $10,000 loan that is borrowed at an annual interest rate of 5.6% for 10 years.

15. Logistic population growth. A food-limited animal population can be described by the function:

$$P(t) = P_0 + (P_f - P_0)\left[\frac{2}{1 + e^{-t/\tau}} - 1\right]$$

Where τ represents a characteristic growth time, P_0 is the initial population, and P_f is the final population. Write a program, `LogisticGrowth.m`, to calculate the population at a specific time t. Pick sensible values for the parameters.

16. Triangulating height. A surveyor who wants to measure the height of a tall tree positions his inclinometer at a distance d from the base of the tree and measures the angle θ between the horizon and the tree's top. The inclinometer rests on a tripod that is 5 feet tall. Write a program, `Triangulate.m`, to calculate the height of the tree. Use reasonable values for d and θ.

17. Resistors in parallel. The total resistance R of two resistors, R_1 and R_2, connected in parallel, is given by:

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$$

Write a program, `Rparallel.m`, to calculate the total resistance R for resistors connected in parallel for each of the following pairs of resistors.

a. $R_1 = 100\ k\Omega$ $\quad R_2 = 100\ k\Omega$

b. $R_1 = 100\ k\Omega$ $\quad R_2 = 1\ \Omega$

c. $R_1 = 100\ k\Omega$ $\quad R_2 = 10\ M\Omega$

18. Compound interest. The value V of an interest-bearing investment of principal, P, after N_y years is given by:

$$V = P\left(1 + \frac{r}{k}\right)^{kN_y}$$

where k is the number of times per year the investment is compounded, and r is the interest rate (5% interest means $r = 0.05$). Write a program, `CompInterest.m`, to calculate the value of such an investment for realistic parameters. Then consider the limiting case of a \$1 investment at 100`%` interest compounded (nearly) continuously with $k = 1 \times 10^9$ for one year. What is the value of the investment after one year in the limiting case? (Do you recognize this number?)

19. **Paint coverage.** A typical latex paint will cover about 400 square feet per gallon of paint. Write a program, `CalcPaint.m`, that determines the number of gallons a consumer should purchase to have at least a minimum amount of paint to apply two coats of paint to a room with a given length, width, and wall height, a given number of windows with specified dimensions and doorways of specified dimensions. Run the program for a $16' \times 20'$ room with ceiling height $8'$ with four $30'' \times 4'$ windows and two $3' \times 7'$ door openings.

or

```
rand('seed',sum(100*clock));
```

Looking ahead

A very common computational task is to represent a mathematical function, or a set of measured data, by enumerating ordered pairs such as $\{(x_1, y_1), (x_2, y_2), (x_3, y_3) \ldots\}$. To visualize this set, you may often want to plot these points, perhaps connected by line segments. The vectors developed in this chapter make it natural to represent a set of ordered pairs like this by a pair of MATLAB vectors `x` and `y`, which have the same length and store the components of the pairs: $x_3 =$ `x(3)` and $y_3 =$ `y(3)`, etc. The next chapter describes how to make plots from pairs of MATLAB vectors. Visualizing the relationship between tabulated sets of data (vectors) is a real strength of MATLAB.

PROGRAMMING PROBLEMS

1. **Greeting.** Write a MATLAB program, `greeting.m`, that asks for the first name of the user and then greets the user. Something like this:

```
What is your first name? Alfonso
Good morning, Alfonso! Glad to be computing with you today.
```

2. **Quadratic roots2.** Write a program, `quadroots2.m`, to prompt the user to enter in turn a, b, and c, and then calculate the roots of the quadratic equation. Something like this:

```
***********************************

Quadratic Solver for ax^2+bx+c=0
Please enter a: xx
Please enter b: xx
Please enter c: xx
The roots are:
        Z1=xxxxx
        Z2=xxxxx
***********************************
```

3. **Pig Latin.** Write a program, `piglatin.m`, prompting the user to enter a string that starts with a consonant to be translated into pig Latin. The rule for pig Latin is the following: move the first letter of the word to the end with an added “ay.” For example, the word “hello” translates into “ellohay.” Translate the user’s word and then display to the user the word and the pig Latin translation. (This program need not handle digraphs like “th” correctly.)

4. **Sum next 10.** Write a program, `sum10.m`, which asks the user for an integer and then finds the sum of that number and the next nine integers following that number. Report the result to the user, which should look something like: `The sum of the integers from 4 through 13 is 85`.

5. **Unit vector.** A unit vector is a vector of length one. It is often desirable to find a unit vector in the same direction as a given vector. Such a vector can be determined by computing $v/|v|$ where $|v| = \sqrt{v_1^2 + v_2^2 + v_3^2 + \cdots + v_n^2}$. Write a program, `unitvector.m`, that finds a unit vector for any four-dimensional vector.

6. **Trig identity.** Write a program, `trigid.m`, that takes an array of five angles and computes two new arrays, the sines of the elements in the first array, and the cosines. Then use array operations to show that $\cos^2(x) + \sin^2(x) = 1$ for each angle in the original array.

7. **Test statistics.** Write a program, `teststats.m`, that prompts the user to enter 5 test scores. Then compute the mean, median, mode, and variance for the scores entered using the appropriate MATLAB functions. (Use `help <function>` or `doc <function>` if you need information on these functions.) Report the results for these test scores to the user.

8. **Box features.** Write a program, `box.m`, that asks the user to input the dimensions of a box and then calculates the surface area and volume of the box. Report the results to the user in the command window.

9. **Poor Encryption.** After registering with a particular website, a student is assigned a 9-character password that contains no numbers. The student needs to email the password to a parent so that the parent has access to the site as well, but is concerned that someone may hack the email and obtain the password. Write a program, `encrypt.m`, that asks the student for the password and a 10-digit telephone number (without dashes). Then use the telephone number to separate characters in the password to create a new 19-character password that the student can email to a parent. The parent can then eliminate the phone numbers in the 19-character password to obtain the correct password.

10. **Better encryption.** The previous problem is clearly rather poor encryption. Can you do better? Write two programs, `myEncrypt.m` and `myDecrypt.m`, that use a different scheme to encrypt a string and decrypt it.

11. **Password.** Write a program, `password.m`, that asks for a user's first name, middle initial, last name, and 10-digit cell phone number. Create a six-character password for the user, using the first letter of the first name, the middle initial, the first two letters of the last name and the last two digits of the telephone number. Report the password to the user. All letters in the password should be lowercase.

12. **Take order.** Write a MATLAB program, `Take Order.m`, which greets the customer, takes the customer's name, and the number of different types of thurgins they want to order. Running the program should look something like this (be careful with spaces):

```
>> TakeOrder
Please enter your name: Buford
Good morning, Buford. We have a variety of thurgins to choose
from today.
How many nurvels would you like? 7
How many tombits? 12
How many weenives? 3
That will be 22 thurgins total.
Thanks for your order, Buford.
```

13. **Take order 2.** Make an expanded version, `TakeOrder2.m`, of the previous program, which also computes the total cost of the order. Say nurvels are \$0.55 each, tombits are \$0.45, and weenives have gone up to \$1.23 each. Print out a full summary of the transaction.

14. **Vector products.** Write a program, `dotcross.m`, that finds the dot product and cross product of two three-dimensional row vectors. Recall that the dot product of two three-dimensional vectors is defined to be $v \cdot w = v_1 w_1 + v_2 w_2 + v_3 w_3$ and the cross product of two vectors is defined to be the vector $v \times w = (v_2 w_3 - v_3 w_2, v_3 w_1 - v_1 w_3, v_1 w_2 - v_2 w_1)$. Display the results in the Command window.

15. **Terminal velocity.** Objects dropped from a great height fall with increasing speed until the drag force from the air balances the force due to gravity. The terminal velocity in air (for sufficiently massive bodies like pennies or people) is given by:

$$v_t = \sqrt{\frac{2mg}{\rho A C_d}}$$

Where m is the mass of the object, g is the acceleration due to gravity ($9.81\, m/s^2$), ρ is the density of air ($1.18\, kg/m^3$), A is the cross vertical sectional area of the object (m^2), and C_d is the dimensionless drag coefficient, which depends on the details of the shape and is about 0.3. Write a program, `TerminalVel.m`, that prompts the user for the m and A, and computes the terminal velocity in m/s and in mi/hr, displaying both in a well-formatted style. Use your program to estimate the difference in speed between a skydiver falling feet-first and lying horizontally.

16. **Squares of numbers.** Write a program, `squares.m`, that uses `linspace` to create and array of integers from 10 to 20 inclusive and then computes and reports the square of each integer.

17. **Squares with user input.** Modify the `squares.m` program to allow a user to enter a range of integer values (e.g., from 15 to 30) and then compute and report the square of each integer. Call the program `squares2.m`.

18. **User paint.** Write a program, `userpaint.m`, which asks the user for the dimensions on the room to be painted, the dimensions of the windows in the room, the number of windows, the dimensions of door openings in the room, and the number of door openings. Compute the number of gallons of paint needed to paint the room, assuming a gallon of paint covers 400 square feet, and report the result to the user. Personalize the program by asking the user for a first name, and report the results with a greeting using the first name.

19. **Take order 3.** Write a new program, `TakeOrder3.m`, which stores the user input of each type of thurgin in an array. Represent the price of each thurgin (as given in the problem above) in a second array and compute the total cost of the thurgins ordered using array operation(s). Finally, report to the user the total cost of the thurgins ordered.

20. **Furniture sales.** A discount furniture store sells four types of bedroom sets. The cost (in dollars) of each set can be represented by the array `Cs = [199, 268, 500, 670]`. The price at which the store sells each set can be represented by the array `Ps = [398, 598, 798, 998]`. In a particular quarter the number of sets sold of each type can be

represented by the array `Ns = [35, 25, 20, 10]`. Write a program, `furniture.m`, that calculates and reports:

a. The total number of bedroom sets sold.

b. The total revenue received by the store from the number of sets sold.

c. The profit realized by the store from the sale of bedroom sets.

21. Rental receipts. Cheap Rentals offers four types of vehicles: compacts; full-sized; vans; SUV's. The daily rental fees for each type of vehicle are $25, $38, $53, and $72, respectively. The following represents their July business, where the number stated is the days rented from each of its three locations: Airport; Campus; Elkhart.

Airport: compacts 250; full-sized 150; vans 180; SUV's 86.

Campus: compacts 160; full-sized 44; vans 60; SUV's 20.

Elkhart: compacts 210; full-sized 112; vans 120; SUV's 78.

Represent the fees and days rented at each location in four appropriate arrays. Then write a program, `rentals.m`, which uses array operations to compute and report:

a. The total number of rental days for July at each location.

b. The total number of rental days for July for the company.

c. The total revenue received at each location in July.

d. The total revenue received by the company in July.