

# Assignment 0

## 1. Distance Traveled

The distance a vehicle travels can be calculated as follows:

$$\text{distance} = \text{speed} * \text{time}$$

For example, if a train travels 40 miles per hour for 3 hours, the distance traveled is 120 miles.

Write a program that asks the user for the speed of a vehicle (in miles per hour) and how many hours it has traveled. The program should then use a loop to display the distance the vehicle has traveled for each hour of that time period. Here is an example of the output:

```
What is the speed of the vehicle in mph? 40
How many hours has it traveled? 3
Hour Distance Traveled
-----
1           40
2           80
3          120
```

*Input Validation: Do not accept a negative number for speed and do not accept any value less than 1 for time traveled.*

## 2. Population Bar Chart

Write a program that produces a bar chart showing the population growth of Prairieville, a small town in the Midwest, at 20-year intervals during the past 100 years. The program should read in the population figures (rounded to the nearest 1,000 people) for 1900, 1920, 1940, 1960, 1980, and 2000 from a file. For each year it should display the date and a bar consisting of one asterisk for each 1,000 people. The data can be found in the `People.txt` file. Here is an example of how the chart might begin:

```
PRAIRIEVILLE POPULATION GROWTH
(each * represents 1,000 people)
1900 **
1920 ****
1940 *****
```

## 3. Lo Shu Magic Square

The Lo Shu Magic Square is a grid with 3 rows and 3 columns shown in Figure 1. The Lo Shu Magic Square has the following properties:

- The grid contains the numbers 1 through 9 exactly.
- The sum of each row, each column, and each diagonal all add up to the same number. This is shown in Figure 2.

In a program you can simulate a magic square using a two-dimensional array. Write a function that accepts a two-dimensional array as an argument, and determines whether the array is a Lo Shu Magic Square. Test the function in a program.

4	9	2
3	5	7
8	1	6

Figure 1

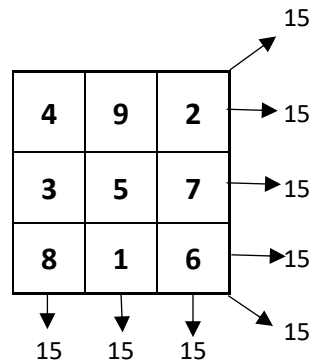


Figure 2

#### 4. Overloaded Hospital

Write a program that computes the charges for a patient's hospital stay and writes the report to a file. First, the program should ask if the patient was admitted as an in-patient or an outpatient. If the patient was an in-patient, the following data should be entered:

- The number of days spent in the hospital
- The daily rate
- Hospital medication charges
- Charges for hospital services (lab tests, etc.)

The program should ask for the following data if the patient was an out-patient:

- Charges for hospital services (lab tests, etc.)
- Hospital medication charges

The program should use two overloaded functions to calculate the total charges. One of the functions should accept arguments for the in-patient data, while the other function accepts arguments for out-patient information. Both functions should return the total charges.

*Input Validation: Do not accept negative numbers for any data.*

#### 5. Quadratic Equation

Write a function solving the quadratic equation  $ax^2 + bx + c = 0$ .

Function *quadraticEquation()* takes the parameters a, b and c and returns the number of roots and the roots (if they exist).

## **6. Median Function**

In statistics, when a set of values is sorted in ascending or descending order, its median is the middle value. If the set contains an even number of values, the median is the mean, or average, of the two middle values. Write a function that accepts as arguments the following:

- a. An array of integers
- b. An integer that indicates the number of elements in the array

The function should determine the median of the array. This value should be returned as a double. (Assume the values in the array are already sorted.)

Demonstrate your pointer prowess by using pointer notation instead of array notation in this function.

## **7. Reverse Array**

Write a function that accepts an int array and the array's size as arguments. The function should create a copy of the array, except that the element values should be reversed in the copy. The function should return a pointer to the new array. Demonstrate the function in a complete program.

## **8. Test Scores**

Write a program that dynamically allocates an array large enough to hold a user-defined number of test scores. Once all the scores are entered, the array should be passed to a function that sorts them in ascending order. Another function should be called that calculates the average score. The program should display the sorted list of scores and averages with appropriate headings. Use pointer notation rather than array notation whenever possible.

*Input Validation: Do not accept negative numbers for test scores.*