ASSIGNMENT 1

1. Distance Traveled

The distance a vehicle travels can be calculated as follows:

*distance = speed * 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. 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)

3. 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.

4. **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.

5. **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.