Sheet 2

5.4 (Conversion from miles to kilometers) Write a program that displays the following table (note 1 mile is 1.609 kilometers):

Miles	Kilometers
1	1.609
2	3.218
. . . . .	
9	14.481
10	16.090

**5.7 (Financial application: compute future tuition) Suppose the tuition for a university is $10,000 this year and increases 5% every year. In one year, the tuition will be $10,500. Write a program that displays the tuition in 10 years, and the total cost of four years' worth of tuition starting after the tenth year

*5.9 (Find the two highest scores) Write a program that prompts the user to enter the number of students and each student's name and score, and finally displays the student with the highest score and the student with the second-highest score. Use the next() method in the Scanner class to read a name rather than using the nextLine() method. Assume that the number of students is at least 2

*6.2 (Sum the digits in an integer) Write a method that computes the sum of the digits in an integer. Use the following method header:

```
public static int sumDigits(long n)
```

For example**, sumDigits(234)** returns **9** (= 2 + 3 + 4).

(Hint: Use the % operator to extract digits and the / operator to remove the extracted digit. For instance, to extract 4 from 234, use 234 % 10 (= 4). To remove 4 from 234, use 234 / 10 (= 23). Use a loop to repeatedly extract and remove the digit until all the digits are extracted. Write a test program that prompts the user to enter an integer then displays the sum of all its digits

**6.3 (Palindrome integer) Write the methods with the following headers:

```
// Return the reversal of an integer, e.g., reverse(456) returns 654
public static int reverse(int number)
// Return true if number is a palindrome
public static boolean isPalindrome(int number)
```

Use the **reverse** method to implement **isPalindrome**. A number is a palindrome if its reversal is the same as itself. Write a test program that prompts the user to enter an integer and reports whether the integer is a palindrome

*6.5 (Sort three numbers) Write a method with the following header to display three numbers in increasing order:

```
public static void displaySortedNumbers( double num1, double num2, double num3)
```

Write a test program that prompts the user to enter three numbers and invokes the method to display them in increasing order.

7.2 (Reverse the numbers entered) Write a program that reads 10 integers then displays them in the reverse of the order in which they were read.

*7.7 (Count single digits) Write a program that generates 100 random integers between 0 and 9 and displays the count for each number. (Hint: Use an array of 10 integers, say counts, to store the counts for the number of 0s, 1s, . . . , 9s.)

7.9 (Find the smallest element) Write a method that finds the smallest element in an array of double values using the following header:

```
public static double min(double[] array)
```

Write a test program that prompts the user to enter 10 numbers, invokes this method to return the minimum value, and displays the minimum value.
Here is a sample run of the program:

```
Enter 10 numbers: 1.9  2.5  3.7  2  1.5  6  3  4  5  2

The minimum number is 1.5
```

*8.2 (Sum the major diagonal in a matrix) Write a method that sums all the numbers in the major diagonal in an n × n matrix of double values using the following header:

```
public static double sumMajorDiagonal(double[][] m)
```

Write a test program that reads a 4-by-4 matrix and displays the sum of all its elements on the major diagonal. Here is a sample run:

```
Enter a 4−by−4 matrix row by row:

 1  2  3  4.0

5   6.5  7  8

9   10  11  12

13  14  15  16

Sum of the elements in the major diagonal is 34.5
```

*8.3 (Sort students on grades) Rewrite Listing 8.2, GradeExam.java, to display the students in increasing order of the number of correct answers.

*8.7 (Points nearest to each other) Listing 8.3 gives a program that finds two points in a two-dimensional space nearest to each other. Revise the program so it finds two points in a three-dimensional space nearest to each other. Use a two-dimensional array to represent the points. Test the program using the following points:

double[][] points = { {−1, 0, 3}, {−1, −1, −1}, {4, 1, 1}, {2, 0.5, 9}, {3.5, 2, −1}, {3, 1.5, 3}, {-1.5, 4, 2}, {5.5, 4, −0.5} };

The formula for computing the distance between two points (x1, y1, z1) and (x2, y2, z2) is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$