Ch12

2. Write a program that takes its input from a file of numbers of type **double** and outputs the average of the numbers in the file to the screen. The file contains nothing but numbers of type **double** separated by blanks and/or line breaks.

3.

   A. Compute the median of a data file. The **median** is the number that has the same number of data elements greater than the number as there are less than the number. For purposes of this problem, you are to assume that the data is sorted (that is, is in increasing order). The median is the middle element of the file if there are an odd number of elements, or is the average of the two middle elements if the file has an even number of elements. You will need to open the file, count the members, close the file and calculate the location of the middle of the file, open the file again (recall the "start over" discussion at the beginning of this chapter), count up to the file entries you need, and calculate the middle.

   B. For a sorted file, a quartile is one of three numbers: The first has one-fourth the data values less than or equal to it, one-fourth the data values between the first and second numbers (up to and including the second number), one-fourth the data points between the second and the third (up to and including the third number), and one-fourth above the third quartile. Find the three quartiles for the data file you used for part a. Note that "one-fourth" means as close to one-fourth as possible.

   Hint: You should recognize that having done part a you have one-third of your job done. (You have the second quartile already.) You also should recognize that you have done almost all the work toward finding the other two quartiles as well.

5. A file contains a series of integers separated by spaces. Write a program that read the integers from the file and checks if they are in the form of an arithmetic series **3, 7, 11, 15, . . ., (3+4*n)**. The series starts with 3 and every next element is 4 greater than the previous one. Your program should print the numbers to an output file, in order of the series, until a number is found that is not a part of the series, or the end of the file is reached.