CS2006 C++ Lab Exercises

1. Largest/Smallest Array Values
   Write a program that lets the user enter 10 values into an array. The program should then display the largest and smallest values stored in the array.

2. Rainfall Statistics
   Write a program that lets the user enter the total rainfall for each of 12 months into an array of doubles. The program should calculate and display the total rainfall for the year, the average monthly rainfall, and the months with the highest and lowest amounts.

3. Nachos and Salsa
   Write a program that lets a maker of nachos and salsa keep track of sales for five different types of salsa: mild, medium, sweet, hot and zesty. The program should use two parallel 5-element arrays: an array of strings that holds the five salsa names and an array of integers that hold the number of jars sold during the past month for each salsa type. The salsa names should be stored using an initialization list at the time the name array is created. The program should prompt the user to enter the number of jars sold for each type. Once this sales data has been entered, the program should produce a report that displays sales for each salsa type, total sales, and the names of the highest selling and lowest selling products.

   Input Validation: Do not accept negative values for number of jars sold.

4. Grade Book
   A teacher has five students who have taken four tests. The teacher uses the following grading scale to assign a letter grade to a student, based on the average of his or her four test scores.

Test Score	**Letter Grade**
90-100	A
80-89	B
70-79	C
60-69	D
0-59	F

   Write a program that uses an array of string objects to hold the five student names, an array of five characters to hold the five students' letter grades, and five arrays of four doubles to hold each student's set of test scores.

   The program should allow the user to enter each student's name and his or her four test scores. It should then calculate and display each student's average test score and a letter grade based on the average.

   Input Validation: Do not accept test scores less than 0 or greater than 100

5. 2D Array Operations
   Write a program that creates a two-dimensional array initialized with test data. Use any data type you wish. The program should have the following functions:
   - getTotal. This function should accept a two-dimensional array as its argument and return the total of all the values in the array.
   - getAverage. This function should accept a two-dimensional array as its argument and return the average of all the values in the array.
   - getRowTotal. This function should accept a two-dimensional array as its first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the total of the values in the specified row.
   - getColumnTotal. This function should accept a two-dimensional array as its first argument and an integer as its second argument. The second argument should be the subscript of a column in the array. The function should return the total of the values in the specified column.
   - getHighestInRow. This function should accept a two-dimensional array as its first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the highest value in the specified row of the array.
   - getLowestInRow. This function should accept a two-dimensional array as its first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the lowest value in the specified row of the array.

   Demonstrate each of the functions in this program.

6. Charge Account Validation
   Write a program that lets the user enter a charge account number. The program should determine if the number is valid by checking for it in the following list:

```
5658845     4520125     7895122     8777541     8451277
1302850     8080152     4562555     5552012     5050552
7825877     1250255     1005231     6545231     3852085
7576651     7881200     4581002
```

   The list of numbers above should be initialized in a single-dimensional array. A simple linear search should be used be used to locate the number entered by the user. If the user enters a number that is in the array, the program should display a message saying that the number is valid. If the user enters a number that is not in the array, the program should display a message indicating that the number is invalid.

7. Write a method

```
boolean sameSet(int a[], int b[])
```

that checks whether two arrays have the same elements in some order, ignoring duplicates. For example, the two arrays

1 4 9 16 9 7 4 9 11

and

11 11 7 9 16 4 1

would be considered identical. Implement this in a program.