**50 Points**

1. Write a program that reads in the average monthly rainfall for a city for each month of the year and then reads in the actual monthly rainfall for each of the previous 12 months. The program then prints out a nicely formatted table showing the rainfall for each of the previous 12 months as well as how much above or below average the rainfall was for each month. The average monthly rainfall is given for the months January, February, and so forth, in order. To obtain the actual rainfall for the previous 12 months, the program first asks what the current month is and then asks for the rainfall figures for the previous 12 months. The output should correctly label the months.   
     
   There are a variety of ways to deal with the month names. One straightforward method is to code the months as integers and then do a conversion before doing the output. A large switch statement is acceptable in an output function. The month input can be handled in any manner you wish, as long as it is relatively easy and pleasant for the user.   
     
   After you have completed the previous program, produce an enhanced version that also outputs a graph showing the average rainfall and the actual rainfall for each of the previous 12 months. The graph should be similar to the one shown in Display 5.4 , except that there should be two bar graphs for each month and they should be labeled as the average rainfall and the rainfall for the most recent month.   
     
   Your program should ask the user whether he or she wants to see the table or the bar graph, and then should display whichever format is requested. Include a loop that allows the user to see either format as often as the user wishes until the user requests that the program end.

**20 Points**

1. Generate a text-based histogram for a quiz given to a class of students. The quiz is graded on a scale from 0 to 5. Write a program that allows the user to enter grades for each student. As the grades are being entered, the program should count, using an array, the number of 0’s, the number of 1’s, the number of 2’s, the number of 3’s, the number of 4’s, and the number of 5’s. The program should be capable of handling an arbitrary number of student grades (stop input when -1 is entered).  
     
   You can do this by making an array of size 6, where each array element is initialized to zero. Whenever a zero is entered, increment the value in the array at index 0. Whenever a one is entered, increment the value in the array at index 1, and so on, up to index 5 of the array.  
     
   Output the histogram count at the end. For example, if the input grades are 3, 0, 1, 3, 3, 5, 5, 4, 5, 4, -1 then the program should output

1 grade(s) of 0

1 grade(s) of 1

0 grade(s) of 2

3 grade(s) of 3

2 grade(s) of 4

3 grade(s) of 5

**30 Points**

1. Write a program to assign passengers seats in an airplane. Assume a small airplane

with seat numbering as follows:

1 A B C D

2 A B C D

3 A B C D

4 A B C D

5 A B C D

6 A B C D

7 A B C D

The program should display the seat pattern, with an 'X' marking the seats already

assigned. For example, after seats 1A, 2B, and 4C are taken, the display should look

like this:

1 X B C D

2 A X C D

3 A B C D

4 A B X D

5 A B C D

6 A B C D

7 A B C D

After displaying the seats available, the program prompts for the seat desired, the

user types in a seat, and then the display of available seats is updated. This continues

until all seats are filled or until the user signals that the program should end.

If the user types in a seat that is already assigned, the program should say that that

seat is occupied and ask for another choice.