COP 3223 Program #8: Pizza Shack is Back!!!

Your week off from Pizza Shack is over. While you enjoyed your excursion into Number Theory, you are happy to be back to use your programming skills to increase happiness of all via improve pizza sales. The first program you'll write will be in the style of a programming contest question. You'll take input from a file and output the relevant data to the screen. The description will be written in the same format of contest questions. The second program is designed to take in data from a weather file taken from the weather archive linked in the lecture notes. ***Your program should work with input from any file from this archive with -1 added to the end.***

Program A: Online Ordering for Pizza Shack (pizza.c)

Pizza Shack would like to allow online ordering for its customers as many other competitors are doing the same thing. For each online order, a customer chooses how many of each size pizza they would like. (For the purposes of making this problem easier, let's just assume that all of the pizzas are just pepperoni pizzas!) Let's also assume that it takes 10 minutes to complete an order from the time the staff starts working on the pizzas, no matter how large the order is. (This also makes the problem easier!)

Your boss would like for you to write a program that takes in the time each order is made, as well as the items in the order and calculate both the cost of the order and the time it will be ready. Keep in mind that an order might not be ready exactly 10 minutes after it's placed, if the staff is busy making pizzas for previous orders. (The staff only works on one order at a time and completes it before moving onto the next one.)

Please use the following constants for the costs of the three sizes of pizza:

```
#define SMALL_COST 6.99
#define MEDIUM_COST 9.99
#define LARGE_COST 12.99
```

and apply 6.5% sales tax to the total of each order.

To simplify time, we will simply output the number of minutes after Pizza Shack has opened for the day. Thus, times that orders will be completed will be positive integers.

Input Format

1. The first line of the input file will contain a single positive integer, *n (n < 100)*, specifying the number of different online orders for Pizza Shack

2. For each order, the first item in the input line will be the number of minutes after the store opens that the order was placed. Each of these numbers will be in between 0 and 1400, inclusive. It is guaranteed that the orders will be such that all of them can be filled by time 1439, 23 hours and 59 minutes after the store opens!

3. The rest of each order line will contain three integers, *S, M* and *L*, separated by spaces, indicating the number of small, medium and large pizzas, respectively, that are part of the order.

Output Format

For each of online orders, output a single line with the following format:

```
Order #k: Ready at time X, cost = $Y.
```

where k is the order number, starting at 1, X is the number of minutes after the store opens that the order will be ready to pick up and Y represents the cost of the order, with tax, rounded to two decimal places, in dollars.

Sample Input File (pizza.txt)

```
5
10 1 2 3
19 5 0 0
22 0 0 9
80 1 1 1
85 0 2 3
```

Sample Output (Corresponding to Sample Input File)

```
Order #1: Ready at time 20, cost = $70.23.
Order #2: Ready at time 30, cost = $37.22.
Order #3: Ready at time 40, cost = $124.51.
Order #4: Ready at time 90, cost = $31.92.
Order #5: Ready at time 100, cost = $62.78.
```

Note: It’s just a random coincidence that all of the times listed are multiples of ten. This would not be true of most test cases.

Program B: Weather Predictions and Pizza (weather.c)

Pizza Shack is looking to open in different locales. After much research, the owner of Pizza Shack has found out that pizza restaurants tend to do better in places with temperate weather. If the percentage of days with an average temperature in between 65 and 75 degrees Fahrenheit is greater than 60%, then the restaurant is likely to succeed.

Of course, it's possible that these three values, the lower temperature bound, upper temperature bound and percentage of days might change over the course of time as people's changes taste.

Your boss has asked you to write a program that takes in the value of those three bounds, as well as historical weather data from the city in which he is thinking about opening a restaurant. Based on this information, output the number of days for which the city had an average temperature in the desired range, the total number of days for which the weather data existed, the corresponding percentage of days with a temperature in the desired range as well as a recommendation of whether or not to build the restaurant!

Input Specification

1. The lower bound temperature in Fahrenheit will be an integer in between 0 and 80.
2. The upper bound temperature in Fahrenheit will be an integer greater than the lower bound and less than 90.
3. The desired percentage will be an integer in between 0 and 100, inclusive.
4. The name of the weather file will be a string less than 20 characters in length and the corresponding file will be in the format of files posted at the website http://academic.udayton.edu/kissock/http/Weather/default.htm, **with a -1 added to the very last line, to make detecting the end of data easier.**

Output Specification

Please output three lines with the following format.

```
Days in between X and Y degrees: A
Total number of days:            B
Percentage of days in range:     C
```

Print he percentage to one decimal place.

If the percentage is greater than or equal to the percentage entered by the user, print out the following fourth line:

```
Great, I recommend opening a Pizza Shack at your location!
```

Otherwise, output the following fourth line:

```
Sorry, your weather isn't temperate enough for Pizza Shack.
```

Sample Program Run (User Input in Bold and Italics)

```
Please enter the lower temperature bound in Fahrenheit.
60
Please enter the upper temperature bound in Fahrenheit.
80
Please enter the percentage of days needed in the range.
60
Please enter the name of the weather data file for your city.
FLORLAND.txt

Days in between 60 and 80 degrees: 4857
Total number of days:              7174
Percentage of days in range:       67.7
Great, I recommend opening a Pizza Shack at your location!
```

Sample Program Run (User Input in Bold and Italics)

```
Please enter the lower temperature bound in Fahrenheit.
40
Please enter the upper temperature bound in Fahrenheit.
60
Please enter the percentage of days needed in the range.
50
Please enter the name of the weather data file for your city.
MABOSTON.txt

Days in between 40 and 60 degrees: 2561
Total number of days:              7178
Percentage of days in range:       35.7
Sorry, your weather isn’t temperate enough for Pizza Shack.
```

Note: These two files are posted in WebCourses. If you download them from the website, you may get a slightly different version with more days added. Also, both of these files have the -1 added as the last token to indicate the end of file. Make sure to add this to any new file you download from the Weather Website.

Deliverables

Please submit two separate **.c files** for your solutions to these problems via WebCourses by the designated due date:

Program A: **pizza.c**
Program B: **weather.c**