Introduction to C - Programming Assignment #2
Driving Decisions!

Due date: *September 22, 2010, 11:59pm*

Note: In this assignment, you are required to write three programs.

Objectives

1. Learn how to write and use `"if"` statements**.**

References

CKnights: Sections 4, 5 GuhaDraft: Chapter 3

Problem A: Fuel Efficiency Rating

You will add to your final version of Program #1 to complete this problem. In particular, you ask the user the same information that you asked them in Program #1, but in addition to printing out the user's fuel efficiency, you will print out a rating for their fuel efficiency. Here is how the ratings are calculated:

Fuel Efficiency, x, (in miles/gallon)	Rating
0 <= x <= 20	poor
20 < x <= 30	average
30 < x <= 40	good
x > 40	excellent

Input Specification

Same exact specifications as Program #1, Part C.

Output Specification

In addition to the output specified in Program #1, Part C, print out a single line with the following format:

```
Your car gets (Rating) gas mileage.
```

Output Sample

Below are three sample outputs of running the program. Note that these samples are NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given above. In the sample runs below, for clarity and ease of reading, the user input is given in *italics* while the program output is in bold. **(When you run your own program, there will be no italics or bold, since text-based output all looks the same.)**

Sample Run #1

```
What is the radius of your tires, in inches?
15
How many revolutions did your car's tires make?
10000
How many gallons of gas did your car use?
0.75
Your car averaged 19.83 miles per gallon.
Your car gets poor gas mileage.
```

Sample Run #2

```
What is the radius of your tires, in inches?
16
How many revolutions did your car's tires make?
3151
How many gallons of gas did your car use?
0.11
Your car averaged 45.45 miles per gallon.
Your car gets excellent gas mileage.
```

Sample Run #3

```
What is the radius of your tires, in inches?
16
How many revolutions did your car's tires make?
3151
How many gallons of gas did your car use?
0.22
Your car averaged 22.73 miles per gallon.
Your car gets average gas mileage.
```

Problem B: Forgetful Fred

Fred is thrilled that he has programmed his car to tell him his average fuel efficiency on each trip. However, this hasn't solved another problem that he has: that he often leaves items at home that he needs for work. Typically, Fred realizes that he has left an important item at home only a few minutes into his drive. At this point, Fred has a difficult decision to make. Should he continue to work without the item, or should he turn around, get the item and then drive to work. Being late to work is very bad (worse than leaving the item), so Fred only opts to turn around and get the item if he can still arrive at work on time. It's stressful for him to make this calculation, so he would like to write a program to make it for him, so that his car can automatically tell him whether or not to turn around.

In particular, your program should prompt the user and read in the following pieces of information:

1) The time it takes to drive to work from home, in minutes.
2) The number of minutes before work starts at the time he/she originally starts the trip.

3) The number of minutes into the drive that the user realizes that the important item was left at home.

With this information, your program should determine whether or not the user should turn around and get the item.

Input Specification

All three input values will be positive integers. The number of minutes before work the user starts the trip will be at least as much as the number of minutes it takes to get to work. In addition, you are guaranteed that the number of minutes into the drive that the user realizes that the important item was left at home will be strictly less than the number of minutes it takes to drive from home to work.

Output Specification

Your first line of output should be one of the following two messages:

```
Go back home and pick up the item.
Just go to work, you'll have to do without the item.
```

The second line of output should specify how many minutes early you will arrive at work (assume you follow the advice about picking up or not picking up the item). This second line of output should have the following format:

```
You will arrive at work with X minutes to spare.
```

where X represents the number of minutes early the user will arrive at work.

Output Samples

Here are three sample outputs of running the program. Note that these samples are NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given above. The user input is given in *italics* while the program output is in bold.

Sample Run #1

```
How long does it take to drive to work (in minutes)?
30
How many minutes before work did you start?
60
How many minutes did it take to realize you forgot an item?
10
Go back home and pick up the item.
You will arrive at work with 10 minutes to spare.
```

Sample Run #2

```
How long does it take to drive to work (in minutes)?
30
How many minutes before work did you start?
60
How many minutes did it take to realize you forgot an item?
20
Just go to work, you'll have to do without the item.
You will arrive at work with 30 minutes to spare.
```

Sample Run #3

```
How long does it take to drive to work (in minutes)?
30
How many minutes before work did you start?
60
How many minutes did it take to realize you forgot an item?
15
Go back home and pick up the item.
You will arrive at work with 0 minutes to spare.
```

Problem C: Forgetful Fred Again

Fred's program works great! Now, he is told whether or not to go home to pick up an item he forgot for work. But, his program isn't good enough. The last time he turned around to pick up the item, his car didn't have enough gas to do the extra driving, and he stalled out on the road on the way back to work after he picked up his item. He realizes that in this situation (where he had enough time, but not enough gas), his program SHOULD HAVE told him to just continue to work, because running out of gas ended up making him late for work anyway! If it turns out that Fred doesn't even have enough gas to make it to work at all, he MUST stop to get gas, which takes an extra five minutes. Fred has realized that there are six possibilities (depending on time and gas) as described below in Output Specification. Adjust your program from part B to handle all these possibilities.

This program will have to ask the user to enter two more pieces of information:
1) The amount of gas left in the car when he/she originally starts to go to work.
2) The amount of gas per minute the car uses when driving.

Input Specification

The last two input values will both be positive real numbers. Also, it is guaranteed that the user will realize that they left the item at home BEFORE they run out of gas and that they will always be able to find a gas station immediately once they realize that they've left the item at home.

Output Specification

There are six possibilities depending on whether or not there is enough time to go back and depending on how much gas there is in the car. To simplify the explanations, these six cases are listed separately below, along with their corresponding output lines. However, you may decide to combine some of these cases in your program; just make sure your output follows the specifications.

Case 1 - There is enough time to go back and there is enough gas to go back and still make it to work:
In this case, print

```
Go back home and pick up the item.
You will arrive at work with X minutes to spare.
```

Case 2 - There is enough time to go back but there is not enough gas to go back and still make it to work (there is only enough gas to make it to work without going back):
In this case, print

```
You don't have enough gas to pick up the item. Go to work.
You will arrive at work with X minutes to spare.
```

Case 3 - There is enough time to go back but there is not enough gas to make it to work regardless of going back or not going back:
In this case, print

```
You need gas. Go to a gas station immediately!
```

Due to the fact that it takes five minutes to go to a gas station, this could conceivably make you late to work (by no more than five minutes). If you are on time, print the second line with the usual format:

```
You will arrive at work with X minutes to spare.
```

Otherwise, print the second line with the following format:

```
You will arrive at work X minutes late.
```

Case 4 - There is not enough time to go back but there is enough gas to go back and still make it to work:
In this case, print

```
You don't have enough time to pick up the item. Go to work.
You will arrive at work with X minutes to spare.
```

Case 5 - There is not enough time to go back and there is not enough gas to go back (there is enough gas to make it to work without going back):
In this case, print

You don't have enough time to pick up the item. Go to work.
You will arrive at work with X minutes to spare.

Case 6 - There is not enough time to go back and there is not enough gas to make it to work regardless of going back or not going back:
In this case, print the following for the first line:

You need gas. Go to a gas station immediately!

Due to the fact that it takes five minutes to go to a gas station, this could conceivably make you late to work (by no more than five minutes). If you are on time, print the second line with the usual format:

You will arrive at work with X minutes to spare.

Otherwise, print the second line with the following format:

You will arrive at work X minutes late.

It is noted that the output for cases 3 and 6 are of the identical format, and the output for cases 4 and 5 are of the identical format.

Output Samples
Here are four sample outputs of running the program. Note that these samples are NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given above. The user input is given in *italics* while the program output is in bold.

Sample Run #1
How long does it take to drive to work (in minutes)?
30
How many minutes before work did you start?
60
How many minutes did it take to realize you forgot an item?
10
How many gallons of gas do you have left?
3.5
How many gallons of gas do you use per minute?
0.05
Go back home and pick up the item.
You will arrive at work with 10 minutes to spare.

Sample Run #2

```
How long does it take to drive to work (in minutes)?
30
How many minutes before work did you start?
60
How many minutes did it take to realize you forgot an item?
10
How many gallons of gas do you have left?
3.5
How many gallons of gas do you use per minute?
0.1
You don't have enough gas to pick up the item. Go to work.
You will arrive at work with 30 minutes to spare.
```

Sample Run #3

```
How long does it take to drive to work (in minutes)?
30
How many minutes before work did you start?
60
How many minutes did it take to realize you forgot an item?
10
How many gallons of gas do you have left?
3.5
How many gallons of gas do you use per minute?
0.2
You need gas. Go to a gas station immediately!
You will arrive at work with 25 minutes to spare.
```

Sample Run #4

```
How long does it take to drive to work (in minutes)?
30
How many minutes before work did you start?
32
How many minutes did it take to realize you forgot an item?
10
How many gallons of gas do you have left?
3.5
How many gallons of gas do you use per minute?
0.1
You don't have enough time to pick up the item. Go to work.
You will arrive at work with 2 minutes to spare.
```

Deliverables

Three source files:

1) *rating.c*, for your solution to Problem A
2) *work.c*, for your solution to Problem B
3) *workfinal.c*, for your solution to Problem C

All files are to be submitted over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using Dev C++. Each of your three programs should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

1) Your correctness
2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.
3) Compatibility to Dev C++ (in Windows). If your program does not compile in this environment, **the maximum credit you will receive is 50%.**