C - Programming Assignment #1
Due date: 02/28/14

Objectives

1. To give students practice at typing in, compiling and running simple programs.
2. To learn how to read in input from the user.
3. To learn how to use assignment statements and arithmetic expressions to make calculations
4. To learn how to use the if-then-else construct.

Problem A: Calculating the Number of Electoral Votes for a State

It is election season and your first task as a volunteer is to write a computer program to determine the number of electoral votes a state has based on its population. Roughly, the number of House of Representative members from a state is determined based on its population plus the number of Senators from a state is two. The sum of these is the number of electoral votes a state has.

In particular, there are 435 members of the House that are proportionally allocated. The US population as of the year 2000 was approximately 299,000,000. Utilize this estimate for your program, since that's what the numbers for this upcoming election are based upon. (Create a constant in your program to store this value.) Now, to determine the number of House members from a state, calculate the following formula:

$$\frac{StatePopulation \times 435}{USPopulation}$$

Since this formula often leads to a non-integral answer, but we must assign an integral number of electoral votes for each state, we will simply assign the integer division of the numerator and denominator of this fraction as the number of House members for a state.

Furthermore, some states have enough people that multiplying their population by 435 will lead to an overflow of an integer. In using this formula, we must use both the state population in thousands and the US population in thousands. (Thus, we would use the number 299,000 to represent the US population and the number 18,000 to represent the population of Florida, since its approximate population was 18,000,000 in the year 2000.)

Note: This formula comes within 1 electoral vote of the actual. Thus, some of your answers may differ from the correct number of electoral votes for a state. For example, using this formula for Florida gives 28 electoral votes, but Florida actually has 27. On the other hand, using the formula yields 2 electoral votes for Alaska, but they have 3 (since each state is required to have at least one member in the House.)

Input Specification

1. The state population will be a positive integer, representing the number of people living in the state.

Output Specification

Output the number of electoral votes for the state as an integer following the output sample provided below.

Output Sample

Below is one sample output of running the program. Note that this sample is 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 run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in bold.

Sample Run #1

```
What is the population of your state?
18000000
Your state has 28 electoral votes.
```

Problem B: Calculating the Number of Electoral Votes to Win

In running a presidential campaign, you realize that there are some states you will definitely win and some states you will definitely lose. Thus, your focus must be placed on the rest of the states. It would be nice to know how what percentage of the electoral votes in the "up for grabs" states you need to win to claim the nomination. In particular, there are a total of 538 electoral votes (Washington DC gets 3) and 270 are needed to win the presidency.

For your program, you must prompt the user for the two pieces of information:

1) The number of electoral votes your candidate has in the bag.
2) The number of electoral votes your candidate will definitely NOT win.

Using these two pieces of information, calculate the percentage of remaining electoral votes your candidate must win to the presidency.

In order to simplify your program, make the two following assumptions:

1) The number of electoral votes your candidate has in the bag will always be fewer than 269.

2) The number of electoral votes your candidate's opponent has in the bag will always be fewer than 269.

These two restrictions guarantee that your candidate has an opportunity to win, but hasn't won yet, so to speak.

Input Specification

1. The number of electoral votes your candidate has in the bag and the other candidate has in the bag will be non-negative integers less than 269.

Output Specification

Output the percentage of the remaining electoral votes needed for your candidate to win to two decimal places. Your output should follow the format below, where XX.XX is percentage in question.

```
XX.XX percent of the remaining vote is needed to win!
```

Output Samples

Below is a sample output of running the program. Note that this sample is 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 run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in bold.

Sample Run #1

```
How many electoral votes does your candidate have already?
100
How many electoral votes does your opponent have already?
135
56.11 percent of the remaining vote is needed to win!
```

Problem C: Which States do you Need to Win?

Many times, a race comes down to two states. This time, you will write a program that assumes that your candidate already has a certain number of electoral votes less than 269, and then, given the number of electoral votes up for grabs in two states, determines which combinations of wins bring victory. (There are three possibilities: winning state one, winning state two, and winning both.)

The following information will be requested from the user:

1) Number of electoral votes already won by your candidate.
2) The name of state one (that is up for grabs)
3) The number of electoral votes in state one
4) The name of state two (that is up for grabs)
5) The number of electoral votes in state two

Input Specification

1. The number of electoral votes won will be a positive integer less than 269.
2. The number of electoral votes for both states will be positive integers less than 56.
3. Both names of states will contain fewer than 20 characters.

Output Specification

For each scenario, output a single line with one of the two following formats:

```
Your candidate wins if he/she wins X.
Your candidate wins if he/she wins X and Y.
```

Here X and Y represent the state names entered by the user. Then, output one statement that summarizes the number of ways of winning (which will be 0, 1, 2 or 3) with the following format.

```
Your candidate can win in W number of ways.
```

Output Samples

Below are two 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 run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in bold.

Sample Run #1

```
How many electoral votes has your candidate won?
235
What is the name of the first state in contention?
Florida
How many electoral votes does it have?
27
What is the name of the second state in contention?
Indiana
How many electoral votes does it have?
11
Your candidate wins if he/she wins Florida and Indiana.
Your candidate can win in 1 number of ways.
```

Sample Run #2

```
How many electoral votes has your candidate won?
260
What is the name of the first state in contention?
Florida
How many electoral votes does it have?
27
What is the name of the second state in contention?
Indiana
How many electoral votes does it have?
11
Your candidate wins if he/she wins Florida.
Your candidate wins if he/she wins Indiana.
Your candidate wins if he/she wins Florida and Indiana.
Your candidate can win in 3 number of ways.
```

Deliverables

Three source files:

1) *votes.c*, for your solution to problem A
2) *votestowin.c* for your solution to problem B
3) *waystowin.c* for your solution to problem C

All files are to be submitted over WebCT.

Note: do NOT submit .exe files.

Restrictions

Although you may use other compilers, your program must compile and run using Codeblocks or 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 Codeblocks or Dev C++. If your program does not compile in this environment, you will get a **sizable** deduction from your grade.