

☰ COSC 102 - Lab B

Debug B and Lab B

Submit both using [Code Assessor](#)

Debug B

[Download Debug B](#)

Download the attached .cpp file, correct the errors and submit.

Lab B

Problem

The US national debt will top \$20 trillion in the near future. Right now, it stands at nearly \$19.5 trillion. This number tends to be so big, that it is beyond human comprehension. For example, \$1 versus \$1,000 is notable even in our minds. However, \$1 billion versus \$1 trillion has less of an impact since the scale is unimaginable.

To help people have some sort of idea, you will develop a program that will translate dollars into distance. To do this, you must know that a dollar is 0.0043 inches thick. So, if we had one-thousand \$1 bills stacked *perfectly* on top of each other, they would stand 4.3 inches tall.

Your program, when completed, will input a distance and a debt and calculate how many trips (back-and-forth) you could make if you stacked the dollars one-by-one on top of each other. One way to think of this is, with \$19.5 trillion in debt, how many times could I stack the dollars to the moon and back? On average, the moon is 238,857 miles away from Earth.

Inputs

The user will input the distance they wish to travel and the amount of money they want to visualize. Use the following prompts when querying for inputs:

"Enter distance (miles): "

"Enter debt (\$): "

Please notice the space after the colon. Also, all inputs will use decimals, so use the appropriate data type!

Process

You will be given distance in miles (M) and debt in dollars (\$). You will need to calculate the number of trips I could make with that distance.

To calculate the number of trips, you first need like units:

`DOLLAR_INCHES = debt x DOLLAR_BILL_THICKNESS_INCHES`

... this will convert dollars into inches. For example, if debt is \$100 and thickness is 0.0043 inches, then \$100 will span 0.43 inches.

`DOLLAR_FEET = DOLLAR_INCHES / 12.0`

$$\text{DOLLAR_MILES} = \text{DOLLAR_FEET} / 5280.0$$

Now, you have converted dollars (\$) into miles (M), so you have like units.

$$\text{TRIPS} = \text{DOLLAR_MILES} / \text{TRAVEL_DISTANCE}$$

In other words, dividing the number of miles that the dollars span by the requested travel distance will yield the number of trips that you could make with the given dollar amount and the given distance.

Outputs

The output must abide by the following rules and look like the sample output provided below:

1. The **debt** must be in a field: **right**-justified, 15 characters wide, precise to 2 decimal places.
2. The **distance** must be in a field: **right**-justified, 25 characters wide, precise to 4 decimal places.
3. The **trips** must be in a field: **left**-justified, 5 characters wide, precise to 0 decimal places (rounded to nearest whole digit, NOT truncated).

```
For a debt of $          XXXX.YY
and a distance of          XXXX.YYYY miles
You can make XXX   trips.
```

Do not print XXXX.YY, XXXX.YYYY, or XXX. These are placeholders for actual debt, distance, and trips (respectively).

***NOTE:** Yes, your output may look a little weird with the field widths, but the point is to get practice using field widths, precision, and justifications.

Additional Requirements

1. You MUST use a constant to store the thickness of a dollar bill. DO NOT hardcode 0.0043 in your int main(). Instead, declare a constant and use the constant throughout your program. [**refer to chapter 2.5 "constants"**]
2. You must have comments which document the logic of your code [**refer to chapter 2.5 "comments"**]
3. Your code must be properly indented and formatted [**refer to chapter 2.5 "indenting"**]

Topics Covered

Advanced arithmetic
I/O manipulators
Constants

Textbook Chapters Covered

Chapter 2.1
Chapter 2.2
Chapter 2.3
Chapter 2.5 (comments, indenting, constants)
Chapter 6.2 (I/O manipulators, setting precision, rounding)

Relevant Reading

More Data
Simple IO
Assignments

