

## **Week #2: Problems for Junior Knights (Python)**

### **Problem A: Picture Storage (Required)**

Write a program that calculates the number of pictures that can be stored on a thumb drive. Ask the user to enter the number of gigabytes of data the thumb drive can store, as well as the length and width of each picture, in pixels. Assume that each pixel takes 3 bytes of storage. (This isn't really the case, since most pictures are stored in a compressed format.) Output your answer as a whole number. (Hint: Use integer division.)

\*Note: There are **1073741824** bytes in a Gigabyte.

### **Sample 1: Input and Output**

How many gigabytes can your thumb drive store?

**4**

What is the width of each picture in pixels?

**3168**

and height?

**4752**

You can store **95** pictures on your thumb drive.

### **Problem B: Changy Money (Required)**

As a frequent world traveler, Arup Guha has to exchange money to get various different currencies. However, he needs some help to determine exactly how much money he will have left after one of his trips. You will write a program to help him with his estimate. In particular, you will prompt the user for the following information:

- 1) How much US currency he/she has.
- 2) The exchange rate from US currency to the foreign currency.
- 3) How much of the foreign currency he/she spent.

The user exchanges all of his/her money for their trip. Usually, however, the user will have some money leftover when his/her trip is done. At this point, the user exchanges the rest of this currency for US dollars. Your goal will be to determine how much money the user has left in US dollars after his/her trip.

The charge for an exchange of currency is \$2 US. Thus, if you are exchanging 1002 US dollars for Canadian dollars and the Canadian exchange rate is 1.5, then you will receive exactly 1500 Canadian dollars. If you spend 597 Canadian dollars, then you have 903 Canadian dollars left to exchange back to US dollars. This evaluates to \$602 US. However, since the exchange fee is \$2 US, the user would actually have exactly \$600 US.

You may assume that all the values the user enters are valid values. Specifically, you may assume that the user enters positive values for all three entries, and that the user will always have more than \$2 US left after their trip is over. (This is necessary to pay for the transaction fee.)

**Note:** For actual exchange rates, go to <http://www.xe.com/ucc/>.

### **Coding & Debugging**

Rather than trying to get the whole thing to run at once, attempt to code up portions of the program one at a time. After finishing a portion, test that portion. After you are confident that this works properly, move on by adding more functionality to your program. This is called incremental development. You develop your program piece by piece, testing & debugging each piece as you go along. When you are finally done developing your program, you must test the entire application. Three sample runs of how your program should work are listed below. On your own, create several more sample inputs to try to test the validity of your program.

### **Sample Program Runs**

(Note: Computer output is in plain text while the user's input is in bold for these examples.)

#### **Sample Run #1**

How many US dollars are you exchanging?

**1002**

What is the exchange rate from US dollars to the foreign currency?

**1.5**

How much foreign currency did you spend?

**597**

You will be left with \$600.00 US currency.

#### **Sample Run #2**

How many US dollars are you exchanging?

**500**

What is the exchange rate from US dollars to the foreign currency?

**35**

How much foreign currency did you spend?

**10000**

You will be left with \$210.29 US currency.

Note: There is no need for you to correctly format the dollar value. You may simply print out the double variable that stores the correct value without specifically formatting it to print only 2 decimal places. (It will print many decimal places by default.)

### **Problem C: Trains (Optional)**

Imagine a two-way straight railway and two trains approaching each other from opposite directions. You need to compute how long (in minutes) would it take for these trains to come side by side. You will also compute how many miles each train travels till then. Write a program to read in the distance (in miles), speeds of the trains (in miles per hour), and output the time to meet (in minutes) and the distance traveled by each train (in miles).

#### **Sample 1: Input and Output**

What is the distance between the two trains (in miles)?

**4**

What is the speed of the first train (in miles per hour)?

**120**

What is the speed of the second train (in miles per hour)?

**80**

The trains will meet in **1.2** minutes.

The first train will travel **2.4** miles and the second train will travel **1.6** miles