

Assignment 1 - Interactive Input and Output, File Input, Expressions, and Defined Constants

Description

You live in the Czech Republic, where the currency is the koruna. The koruna, like most currencies, is valued against the dollar differently over time. You find that on January 1 in one particular year that 42 korunas are worth one U.S. dollar. A year later you discover that one U.S. dollar is now worth 37 korunas. Write an interactive program that prompts the user for how many korunas the user has in a savings account, and calculates the difference in the number of U.S. dollars between the first year's rate and the second year's rate.

In this assignment, you develop a program to convert currency from korunas to dollars at two different exchange rates. This program is to be written so that input is interactive (obtained from the user), and some of the output (the "report") is printed directly to a disk file (see specs below for naming convention).

Specifications

The interactive portion of the program (which appears on the screen) should follow the following sample dialog (the italicized value of 30000 is a sample input value) [there is no blank line before the *** line]:

```
*** Koruna Exchange App ***
```

```
How many korunas do you have in your savings account? 30000
```

```
The exchange information for 30000 korunas is now being recorded.
```

The "recorded" output (that which is sent to the disk file) should appear exactly as specified below (only the values will change based on what was input during the interactive portion). Note that in the interactive portion, you will assume whole korunas (no decimals) and will display 30000 as 30000 (not 30000.00).

The "recorded" output for this particular amount should appear as follows (there is no blank line at the top of the file):

```
For 30000 korunas:
At the rate of 42 korunas per U.S. dollar,
  you have 714.29 U.S. dollars.
At the rate of 37 korunas per U.S. dollar,
  you have 810.81 U.S. dollars.

The difference is 96.53 U.S. dollars.
```

Note that there should not be any extra leading spaces in front of the dollar amounts. Also, do not assume that the first koruna rate is necessarily larger than the second.

The difference should appear always as a positive number. To do this, you will need to determine the absolute value of the difference. Absolute value of a number is that same number expressed as a positive number (in other words, it gets rid of the negative sign). So the absolute value of -3.8 is 3.8 and the absolute value of 4.5 is 4.5. You will need the absolute value function for float numbers. Remember how we did the **floor()** function in class? How it used a single argument in the parentheses and returned an integer to the expression in which it was used? Well, now you will be using the floating point absolute value function; it is called **fabs()** and has a single float argument. The float value it returns is the absolute value of the argument. It is in the **math.h** library file so think about what you need to do at the top of your program to use it. And there are a couple of ways you can use **fabs()**, depending on how you write your program. To get comfortable with it, write a little baby program that inputs a number from the user and prints the absolute value. Try that program with several positive and negative values.

Please note the line spacing above. There is a blank line between the first 5 lines and the "difference" line. Incorrect spacing will reduce your score.

For full credit, you should use defined constants to represent the two exchange rates throughout the entire program (including the output statements). Do not read the exchange rates in as input values. You may assume these exchange rates are always integer values (not real life, but I'm trying to make it easy for your first program!)

And keep it simple; let the computer do any "math" necessary.

Be sure to include appropriate comments and indentation in your source code. Follow the examples in the text and the sample programs that include comments.

Do not change the specs of this program or your grade will suffer! An important lesson to be learned in programming is to follow the specs exactly. So, for instance, do NOT print the report section of the output to both the screen and to the file in your final submitted version. Don't change the wording or the spacing of the lines. I want you to be meticulous!

Important Warning

Do **not** use tabs or the `\t` character in your program source code or in your output statements. They will mess up your alignment when run on a different computer. Use the space bar to space things out in both your output statements and when typing your source code. When you originally set up your C environment, I had you disable tabs and to not keep trailing spaces. If you didn't do this, please review the appropriate section in Module 2.

This is critical, or your source code will be improperly spaced and lined up. And your program may not work properly with a text data file. If you make sure these are set properly, you can then use the tab key on your keyboard. It will not generate a tab character though; it will generate spaces as if you had pressed the space bar.

What You Are Expected To Submit

For this assignment, I want you to develop both a pseudocode design for the program, and the C program itself. You will not be turning in the pseudocode design. You WILL turn in the C source code file (the **.c** file) and the output disk file output (the **.txt** file).

Your source code external filename should be YOUR first initial and last name and the addition **-ke** (meaning **koruna exchange**) plus the **.c** extension (as an example, it would be **kjefferies-ke.c** for me). Please also make sure that you submit the source code file with the extension of **.c**, not **.cpp**.

Your output text file external filename should be YOUR first initial and last name and the addition **-ke** (meaning **koruna exchange**) plus the **.txt** extension (as an example, it would be **kjefferies-ke.txt** for me). Your program should write this file directly to the **c:\class** folder on your hard drive. You will need to create this folder (all lower-case letters) PRIOR to developing your program. You may do so via Windows Explorer, or by going to the Windows command prompt and using the **md** command (e.g. **md c:\class** then pressing the enter key). [Note: This step is required because Windows does not allow writing directly to the root folder of C:\.] So in the end, your program will create a file called similarly to: **c:\class\kjefferies-ke.txt** (using YOUR name as described above). Also, remember there's something special you need to do in the **open()** command in your program in terms of specifying the filename.

Please submit BOTH the source code file and the output disk file in the assignment submission area of the assignment posting by uploading them. Do not submit the **.exe** file (the "executable").

Make sure you have uploaded BOTH files prior to pressing Submit in the submission area. Please follow the same basic steps you saw in the How to Submit video for Assignment 0.

Please do not submit any extra files other than the ones I listed above. Also, please follow the naming conventions dictated above. Failure to do so will cause you to lose points.

To test your program, run it at least with two different sets of input data: once with 30000 for the input value and once with 5000 as the input value. Submit only the report file from the run which uses 30000. Remember, with each run only the data should change; the program should not at all have to be altered for both data values to work correctly. Check your answers carefully! Important: the program should work for ANY whole koruna value from 0 up to 10000000. For numeric values, use only the data types discussed in the lecture materials (int, float), and be certain you use the correct types! Review what range of values are appropriate for each type and for your particular version of Dev C++ (32 vs 64 bit).

Notes

This is your first C program, so please do not panic! At the same time, I (like your boss) will be picky about output spacing and following the specs, so make sure you do so. If there are any questions, please use Ask Your Instructor a Question in Canvas. Please work independently from other students, and please do not be tempted to subvert your knowledge by cheating.

Best of luck, and have fun!

modified 1/24/2018