CS 150 Lecture 12 Exercises

Complete each of the exercises below and upload to Canvas before the deadline.

A. Data Loops & Accumulating Algorithms

Write the function **sumEvens()** that adds all of the even numbers inside its **istream** argument, using a **data loop**. Paste the requested information in the provided areas below.

|  |
| --- |
| *Copy and paste a screenshot of the source code for your function* |

|  |
| --- |
| *Copy and paste a screenshot of the function tests* |

B. Process Filters - The CS 150 Encryption Library

An **encryption program** scrambles the bytes in a file so that the file is unreadable except to those who know the decryption method and the secret keyword. Ignoring 2,000 years of progress in the field of encryption, we will use a method familiar to Julius Caesar, replacing A with a D, B with an E, and so on



Complete and test the **cipher()** and **plaintext()** functions in the **encryptlib** library.

|  |
| --- |
| *Copy and paste a screenshot of the source code for the cipher function* |

|  |
| --- |
| *Copy and paste a screenshot of the source code for the plaintext function* |

|  |
| --- |
| *Copy and paste a screenshot after running the unit tests* |

C. Files and Line-Oriented I/O

To read a line of text (instead of a character) we use the string function **getline()**. Write a function named **flipLines()** that accepts the name of a file as its parameter. The function opens and writes to the console the same file's contents with successive pairs of lines reversed in order. If the file cannot be opened, then "Cannot open filename.txt" is printed to **cerr**, with **filename.txt** replaced with the actual name.

For example, if the input file contains the following text:

Twas brillig and the slithy toves  
did gyre and gimble in the wabe.  
All mimsey were the borogroves,  
and the mome raths outgrabe.  
  
"Beware the Jabberwock, my son,  
the jaws that bite, the claws that catch,  
Beware the JubJub bird and shun  
the frumious bandersnatch."

The program should print the first pair of lines in reverse order, then the second pair in reverse order, then the third pair in reverse order, and so on. Therefore your method should produce the following output to the console:

did gyre and gimble in the wabe.  
Twas brillig and the slithy toves  
and the mome raths outgrabe.  
All mimsey were the borogroves,  
"Beware the Jabberwock, my son,  
  
Beware the JubJub bird and shun  
the jaws that bite, the claws that catch,  
the frumious bandersnatch."

Notice that a line can be blank, as in the third pair. Also notice that an input file can have an odd number of lines, as in the one above, in which case the last line is printed in its original position.

|  |
| --- |
| *Copy and paste a screenshot of testing output here.* |

|  |
| --- |
| *Copy and paste a screenshot of the source code for the flipLines function* |

D. File Input and Output

Complete the function **processExpenses()** in the file **expenses.cpp**. Open the input and output files and print error messages if they cannot be opened. Each line in input contains the name of a person. The name may consist of one or more words. The name is followed by a list of expenses for that person. Each expense may have a description, but it doesn't have to.

Your job is to:

* Open the input file. Display an error message if it can't be opened.
* Print name of each person, then their total expenses, one per line
* Print your output to the output file.

Here are two lines from the file.

Stacey Jasminka Hyacinthus Mia 25609.10 10957.10 3294.50 5309.70 35630.10 4292.20

Vivianus Stefanija 26185.50 7401.90 (beer) 16506.60

Use single-character input, formatted input and some auxiliary input methods. Run and shoot the screenshots requested below.

|  |
| --- |
| *Copy and paste a screenshot of the program tests* |

|  |
| --- |
| *Copy and paste a screenshot of the expenses.cpp source code.* |