Bubba Hotep Moving and Storage, Inc. (BHMSI) - *Bulk Data Input*

Overview

Your application proved to be a great success and resulted in BHMSI approving the next phase of development.

Customer counts have increased to the point that keyboard entry of order information is no longer realistic. Now, prospective customers enter their move data on Bubba's web site. Each day the web data is consolidated in a text file at corporate headquarters. Our task is to develop an application that will read the file of customer data, validate entries, perform calculations, and generate summary output reports.

File Processing

Your software shall prompt the user to enter the name and the path to the file containing customer data. Once that information is entered, your software must open the file and process the input data. If the input file fails to open, the software must output an error message indicating a problem occurred opening the file and a notice that processing cannot continue. Abnormal exits are not allowed; use `if/else` structure to ensure that code does not execute if the file fails to open.

If the file opens successfully, the software must read and process each line, or record, in the file. The first line of the file contains column headings. These are for anyone reading the file manually. The program should read the entire line of headings to "get it out of the way" and then move on to the second line. The second line of the file is the first record of data that we will process. We will not know how many total records are in the file. We simply shall continue reading and processing lines of data until reaching the end of the input file. Each line of the file contains the following data elements:

Estimate Number	Unique ID (not really a number, but a string, no spaces)
Estimate Date	Date customer entered data (string with format yyyy/mm/dd)
Move Date	When customer wants to move (string with format yyyy/mm/dd)
Move Type	One of 3 choices (a single character)
Distance	Driving distance from origin to destination, in miles (integer)
Weight	Weight of contents to be moved, in pounds (integer)
Pianos	Count of pianos to be moved (integer)
Stairs at origin	More than 15 stairs at the origin (a single character)
Stairs at destination	More than 15 stairs at the origin (a single character)
Customer name & email	Customer's full name and email (string **with** spaces)

Data Validation

The system must test for a variety of possible data errors. You do not need to identify *data type* errors. That means that if a number is expected, then you may assume that the input data file will have a number. If a string is expected, you may assume the input data file will have a string, etc. The possibility of extraneous characters being present in the data file does not apply. Once you have extracted the value of one data element from the file, you may assume what follows is the value of the next data element.

As previously mentioned, if the input data file fails to open, then no calculations will be made, and no output will be sent to the terminal screen. Otherwise, for this version of the project, validation errors will **not** cause the processing to stop. If a data validation check fails, then no calculations will occur for ***that*** record. However, the values of that row of data will be output along with a list of the errors that were identified (see the sample output

presented later). The table below lists BHMSI's business rules (some from Project #1, some new) that apply to data validation checks and other calculations. Data items not listed require no validation.

Data Item	Validation Rule(s)
Estimate Date	Data entry date, will be in the format yyyy/mm/dd (see notes below).
Move Date	Move date, will be in the format yyyy/mm/dd (see notes below). Excluding the month of the estimate and the month of the move, there must be at least one complete month between the estimate date and the move date. Excluding the month of the estimate and the month of the move, there must be no more than five complete months between the estimate date and the move date. Dates that do not conform to these rules are errors (the format does not need to be validated, just the number of complete months between the two dates).
Move Type	Must be one of the valid character codes: `B, b, W, w, C, c`. All other values are errors.
Distance	The miles between origin and destination. As before the minimum distance is 1 mile, the maximum distance is 3200 miles. Entries less than 1 are adjusted to 1. No notice of the adjustment is given. Entries over 3200 miles are errors.
Weight	The total weight in pounds to be moved. The minimum weight allowed is 1200 pounds or twice the distance of the move, whichever is greater. The maximum weight allowed is 18000 pounds. Entries less than the calculated minimum or greater than the maximum of 18000 pounds are errors.
Pianos	The number of pianos to move. The minimum is 0, the maximum is 3. Entries below 0 are adjusted to 0. No notice is given. Entries above 3 are errors.
> 15 stairs at origin	Must be one of the following characters: `Y, y, N, n`. All other values are errors.
> 15 stairs at destination	Must be one of the following characters: `Y, y, N, n`. All other values are errors.

IMPORTANT NOTES: When reading a date from the file, your software shall separate the year, month, and day values and store them in integer variables; this applies to the Estimate Date and the Move Date. The year will always have four digits. With respect to the month, the format "mm" means two digits or one digit depending on the month. Similarly, the format "dd" could mean one digit or two digits.

Cost Data

Possible types of moves	services	Price/pound for packing and loading	Price per pound per mile
Basic	Just pick up and deliver, client must pack (except big furniture, BHMSI does that when loading)	$0.065	$0.0011
Wall Pack	Basic plus BHMSI packs paintings and mirrors	$0.256	$0.0011
Complete	BHMSI packs everything	$0.459	$0.0012

If there is a set of stairs having more than 15 stairs, then there is a $132.00 charge. This applies to each such set of stairs (at origin, or destination, or both).

There is a $275.00 surcharge for each piano if there are no stairs at either end of the move. There is a $575.00 surcharge for each piano if there are more than 15 stairs at either origin or destination. There is a $1075.00 surcharge for each piano if there are more than 15 stairs at both origin and destination.

Program Output

When the program runs, it shall always output selected values from the input data file. If there are no errors then the program shall output additional information. For all input records, output everything except the person's name and email and the estimate number. If a record is correct, also output the move estimate number and the total cost of the move. If a record contains errors (failed validation tests), then output error message(s) describing each failed validation test. Following the error message(s) output the customer's name and email address so that they can be contacted to resolve the issues with their estimate information.

Maintain counters of the total records processed, total records with errors, and total records without errors. For records without errors, also keep counters and running totals for each type of move. Counters are the total count of each type of move. Running totals are of the distance, the weight, and the total cost by move type. This will help BHMSI identify services that are most profitable. Output the running totals in a summary table after all records have been processed.

Sample output from an execution of the program:

```
   Order       Move    Move      Dist.    Weight   Number    Stairs   Move         Total
   Date        Date    Code    (miles)   (pounds)  Pianos    O    D   Number       Cost ($)
-------------------------------------------------------------------------------------------
2019/02/13  2019/04/23  b       2198      15908      2       N    Y  BH-1437591    40046.38
2019/02/06  2019/06/08  B        873       9511      1       Y    N  BH-1431009    10026.63

■■■

2018/12/22  2019/04/23  B        691      12334      1       N    n  BH-1447549    10451.78
2018/09/21  2019/08/05  w       3341      22820      4       n    *
       ERROR: the move date is too long after the estimate date
       ERROR: the distance of this move is over the maximum allowed
       ERROR: the weight of items moved is over the maximum allowed
       ERROR: the number of pianos moved is over the maximum allowed
       ERROR: the answer for stairs > 15 at destination is not valid
       Contact customer to resolve issues:  Emily B Moore emily.moore@colorado.edu

2019/01/07  2019/05/19  w        374      10389      3       Y    y  BH-1421876     7758.62

■■■

2019/02/14  2019/08/12  w       1745       8801      1       y    n  BH-1414379    19421.58
2019/01/04  2019/05/07  w       1989      10910      1       y    y  BH-1411773    26937.95


Total Records = 488  Records with Errors = 44  Records without Errors = 444

                TOTALS (records without errors)

       Type        Count    Distance     Weight        Cost
       ---------------------------------------------------------
       Basic         127      215640     1336103   $  2701523.50
       Wall Pack     158      251940     1661598   $  3588913.99
       Complete      159      243230     1685170   $  4003714.31
```

Academic Integrity

This is an individual project and all work must be your own. Refer to the guidelines specified in the *Academic Honesty* section of this course syllabus or contact me if you have any questions.

Include the following comments at the start of your source code file:

```
/*
 *  main.cpp
 *
 *  COSC 051 Spring 2019
 *  Project #2
 *
 *  Due on: February 26, 2019
 *  Author: <your netID>
 *
 *
 *  In accordance with the class policies and Georgetown's
 *  Honor Code, I certify that, with the exception of the
 *  class resources and those items noted below, I have neither
 *  given nor received any assistance on this project.
 *
 *  References not otherwise commented within the program source code.
 *  Note that you should not mention any help from the TAs, the professor,
 *  or any code taken from the class textbooks.
 */
```

These comments must appear **exactly** as shown above. The only difference will be values that you replace where there are "place holders" within angle brackets such as `<netID>` which should be replaced by your own `netID`. For example, I would replace `<netID>` on the "Author:" line with `waw23`.

Submission Details

Post to Canvas a `.zip` file containing your source code and the given Makefile. Locate the assignment `Project 2` on Canvas and attach/upload your file. Do **not** post your executable file. You should ensure that your source file compiles on the server and that the executable file runs and produces the correct output. Use the following file name for your file: `submit.zip`. The code part of this project is due by end-of-day (11:59pm) on February 26th. Late submissions will be penalized 2.5% for each 15 minutes late. If you are over 10 hours late you may turn in the project to receive feedback but the grade will be zero. In general requests for extensions will not be considered. The value for this project is 100 points.

Programming Skills

The programming skills required to complete this assignment include:

- Screen output (`cout`)
- Keyboard input (`cin`)
- Basic data validation
- Basic output formatting
- Basic calculations
- **File input/output**
- **Control structures for repetition**
- **Advanced output formatting**
- **Tabulated output**
- **Advanced data validation**

Optional Milestones

For this project several milestones are provided. You are NOT required to turn anything in or to meet these milestones. Make sure that your code compiles and runs prior to moving on to the next milestone.

Milestone 1 – NLT February 14th

- Create an empty source code file; insert heading comments (copy and paste from Canvas, edit as applicable), add preprocessor directives, add `using namespace std;`
- Add a "skeleton" of function `main()`
- Compile and run (nothing much should happen, this is just a check for compiler errors)
- Add global constants that you will reuse from Project 1
- Add data input prompt statement and input statement for the path/filename of the data file
- Add variable declarations and initializations that you will reuse from Project 1
- Compile and run after each addition

Milestone 2 – NLT February 16th

- Add code to open the data file and test for success, if the file fails to open output an error message but do not attempt to read the file, if the file does open it is good to also output a "success" message
- Add a loop to read each line of the file, **this is just a test**, read the entire line into one string and then output that string to the terminal, once all records have been read, close the file

Milestone 3 – NLT February 20th

- Now, add code to read each line, but extract the individual data fields in the correct variables (instead of one long string)
- Add data validation code (much of this can be copied from the previous project, except for the abnormal exits that are not allowed this time)
- Add detailed output code, includes values from each row read from the file, and error messages for any records that did not pass validation
- Add code to make calculations and output estimate number and cost for records without errors
- Add optional diagnostic output (it really is optional, but could be greatly helpful)

Milestone 4 – NLT February 23rd

- Add code to keep running totals and calculate summary statistics
- Add formatted summary output

Milestone 5 – NLT February 25th

- Fine tune calculations and output
- Ensure program compiles and runs on the server
- Be finished early, Relax and submit code on time!

Grading

This graded assignment is worth 100 points and will be counted as part of the *Programming Projects* category for the course. A grade rubric will be published and distributed separately.

Course Materials Notice

The materials used in Georgetown University courses ("Course Materials") generally represent the intellectual property of course instructors which may not be disseminated or reproduced in any form for public distribution (e.g., sale, exchange, etc.) without the written permission of the course instructor. Course Materials include all written or electronic documents and materials, including syllabi, current and past examination questions/answers, and presentations such as lectures, videos, PowerPoints, etc., provided by a course instructor. Course Materials may only be used by students enrolled in the course for academic (course-related) purposes.

Published course readings (book chapters, articles, reports, etc.) available in Canvas are copyrighted material. These works are made available to students through licensed databases or fair use. They are protected by copyright law, and may not be further disseminated or reproduced in any form for distribution (e.g., uploading to websites, sale, exchange, etc.) without permission of the copyright owner.

More information about intellectual property and copyright can be found here:

https://www.library.georgetown.edu/copyright.

More information about computer acceptable use policy and intellectual property can be found here:

https://security.georgetown.edu/it-policies-procedures/computer-systems-aup