

Programming Assignment #1
Calculation of Natural Gas Bill

Background:

Monthly natural gas billing amounts for a certain city are calculated using various quantities and rates which are available on the billing statement. Several related terms and charges are defined below:

- Basic Charge** = \$8.00 (this fee is charged regardless of the energy usage)
Number of days in the billing period = **Final Date - Initial Date** (from 26 to 35 days)
Initial meter reading, Final meter reading = 5 digit meter readings in CCF (100 cubic feet)
Gas Used = **Final meter reading - Initial meter reading** (add 100000 if negative) - see note below
Gas Cost = **(Gas Used)*(93¢/CCF)** = cost for the gas (does not include tax or Basic Charge)
Utility Tax = \$2.40 (this fee is charged regardless of the energy usage)
Late Penalty = 1.5% of the previous month's bill (before taxes) if it was paid late
Average Daily Cost = **Bill Amount before taxes and not including late penalties**
Number of days in the billing period

Note: the reading on the meter displays 5 digits, so the meter reads counts 99998, 99999, 00000, etc. So if the final meter reading is less than the initial reading (or if the Gas Used is negative), 100000 must be added.

Example:

Initial Date: May 7
Final Date: June 5
Number of days in the billing period = 29
Previous month's Bill: \$62.67 (paid late)
Initial meter reading = 01699 CCF
Final meter reading = 01772 CCF
Gas Used = Final meter reading - Initial meter reading = 01772 - 01699 = 73 CCF

Cost summary:	Basic Charge	\$ 8.00
	Gas Cost = (73 CCF)(93¢/CCF) =	67.89
	Subtotal	\$75.89
	Late Payment Charge (\$62.67)(0.015) =	.94
	Utility Tax	2.40
	Total Current Charges	\$79.23
	Average daily cost (\$75.89/29 days)	\$ 2.62

Program Requirements:

1. The program should display a descriptive program title initially and then request the following information from the user:
 - Initial Date (Month and Day - for example, enter 6 7 for June 7)
 - Final Date
 - Initial meter reading (no leading 0's or they will be read as octal values!)
 - Final meter reading
 - Was last month's bill paid late? If so, enter the amount.
2. Check for improper inputs. If an improper input is entered, display an error message and terminate the program (**return 1** in the main program). Be sure to check for the following:
 - a valid date must be entered
 - initial and final meter readings must be between 00000 and 99999
 - if the user enters the amount for the last month's bill, it must be positive
 - the number of days in the billing cycle must be from 26 to 35
3. The output should display the following information as a minimum (with appropriate units and formatting):
 - Initial Date
 - Final Date
 - Number of days in the billing period
 - Initial meter reading (leading 0's need not be displayed)
 - Final meter reading (leading 0's need not be displayed)
 - Gas Used
 - Amount of previous bill (only if paid late)
 - Late charge (if applicable)
 - Utility tax
 - Cost before tax
 - Total current charges
 - Average daily cost
5. Run your program for at least the following cases and include printed results for each case in your report.

Initial meter reading (CCF)	Final meter reading (CCF)	Initial Date	Final Date	Last month's bill late?	If late, amount was
01699	01772	May 7	June 5	Yes	\$62.67
99956	00088	October 2	October 30	No	
99912	00001	June 30	August 2	No	
76990	77599	February 8	March 9	Yes	\$53.25

Note: Case 1 above is the same example as shown on page 1. Your results should match.

Extra Credit Suggestions:

You can earn up to 10 additional points on this program's grade. The number of points awarded depends on the complexity or creativity of the feature. Here are a few ideas:

1. When determining average daily cost during February, the program as specified above assumes 28 days during February. Modify the program so that the user enters the year also. If the year is a leap year, divide by 29 instead of 28. Run case 4 on the previous page twice: Once for February 2008 and once for February 2009.
2. If an incorrect entry is made by the user, display an error message and request that the user re-enter the information. This can be done fairly easily using a **while loop** (covered in Ch. 6).
3. Allow the user to enter the meter readings with leading zeros. One way to do this is to read the five digits as 5 characters and then convert each character to an integer using the function **atoi(x)** in the **stdlib** library.
4. When you display the meter readings in your final results, display leading zeros.
5. Use your imagination!

Report

Follow the guidelines in the handout Format for Programming Projects (available on the instructor's web page).