Major Obstacles

The main problem that I had to overcome while doing this project was accounting for the 2-cent (June through September inclusive) or 3-cent (October through May inclusive) charge, that is imposed when the number of text messages is greater than 200 and less than or equal to 400. This was only a problem when the number of text messages was greater than 400 because in this case, the customer would not be charged for the previous texts in 201-400 inclusive range because my “if” statement that deals with this issue only runs when the number of texts falls within this range. To solve the problem, when texts were greater than 400, I added $4.00 on to the total if the texts were sent in months June through September inclusive or $6.00 to the total if the texts were sent in months October through May inclusive. By doing this, I accounted for the 2 or 3-cent charge that must be enforced for the 200 texts within the 201 to 400 inclusive range. I obtained $4.00 by multiplying the 2 cents charge by 200. In the same way, I multiplied 3 cents by 200 to obtain the $6.00 figure.

Test Cases

Each of these tests should be performed at months 1, 6, 9, and 12 because these are borderline months and you must make sure that the customer is being subjected to the correct monthly charge if necessary. In addition, each test should be run on a month that is between 1 and 6, 6 and 9, and 9 and 12 to make sure that the proper charge is applied to the phone bill within these months as well.

When minutes are 0: (0 minutes, # texts, name, month #)

Purpose: To make sure the program runs properly when zero is inputted as the value for minutes used. The customer should not be charged at all for minutes.

When minutes are within the range of free:

(499 minutes, # texts, name, month #)

Purpose: To make sure the customer is not charged at all for minutes when the number of minutes is less than 500.

When minutes are on the 500 borderline:

(500 minutes, # texts, name, month #)

Purpose: To make sure the customer is not charged at all for minutes when the number of minutes is exactly equal to 500.

When minutes are no longer free:

(501 minutes, # texts, name, month #)

Purpose: To make sure that the customer is charged the appropriate 45 cents for each text that exceeds 500.

\*Note: the following text test cases are tested at 0, 499, 500, and 501 minutes to make sure that the “if” statements run properly when used in conjunction with each other for all possible scenarios.

No texts sent: (0 minutes, 0 texts, name, month #)

(499 minutes, 0 texts, name, month #)

(500 minutes, 0 texts, name, month #)

(501 minutes, 0 texts, name, month #)

Purpose: To make sure to code runs properly when the input for number of text messages is 0. The customer should not be charged at all for texts.

Texts that are subject to the 2-cent or 3-cent monthly charge but are less than 400:

(0 minutes, 201 texts, name, month #)

(499 minutes, 201 texts, name, month #)

(500 minutes, 201 texts, name, month #)

(501 minutes, 201 texts, name, month #)

Purpose: to make sure that the correct monthly charge is enforced in all cases where a 2-cent or 3-cent monthly charge is necessary.

\*Note that the case of 400 texts is also subject to the 2-cent or 3-cent charge, those test cases are listed later.

Texts on the borderline that are free: (0 minutes, 200 texts, name, month #)

(499 minutes, 200 texts, name, month #)

(500 minutes, 200 texts, name, month #)

(501 minutes, 200 texts, name, month #)

Purpose: to make sure that texts on the 200 borderline do not result in a charge for the customer.

Texts within the range of free: (0 minutes, 199 texts, name, month #)

(499 minutes, 199 texts, name, month #)

(500 minutes, 199 texts, name, month #)

(501 minutes, 199 texts, name, month #)

Purpose: To make sure that the customer is not charged at all for text messages.

Texts on the borderline of the 2-cent or 3-cent monthly charge:

(0 minutes, 400 texts, name, month #)

(499 minutes, 400 texts, name, month #)

(500 minutes, 400 texts, name, month #)

(501 minutes, 400 texts, name, month #)

Purpose: To make sure that the customer is not charged 11 cents for the 400th text and that they are still subject to the 2-cent or 3-cent charge.

Texts above 400 that are subjected to the 11cent charge:

(0 minutes, 401 texts, name, month #)

(499 minutes, 401 texts, name, month #)

(500 minutes, 401 texts, name, month #)

(501 minutes, 401 texts, name, month #)

Purpose: To make sure that 11 cent charge that is imposed when texts are greater than 400 is enforced, while simultaneously making sure that the 2-cent or 3-cent charges for texts between 201 and 400 inclusive were enforced.

The following tests serve to make sure that the correct error message is produced when the user gives erroneous input.

When the number of minutes used is negative: (-# minutes, # texts, name, month #)

When the number of text messages is negative:( # minutes, -# texts, name, month #)

When both minutes and texts are negative: (-# minutes, -# texts, name, month #)

If an empty string is provided for the customers name:

(# minutes, # texts, “”, month #)

\*Along with other variations that have previous errors to make that the only error message that is produced is for the first erroneous input.

If the number is not an integer between 1 and 12 inclusive:

(# minutes, # texts, name, month 0)

(# minutes, # texts, name, month -1)

(# minutes, # texts, name, month 13)

\*Along with other variations that have previous errors to make that the only error message that is produced is for the first erroneous input.

Cases program does not handle correctly:

When the number inputted by the user is too big to be held in an int variable.

Also when a character or string is read in where an integer is needed, but that will not be tested according to the spec.