Carson Roscoe January 18th, 2016

COMP4981 Test Document

Tests Summary

Screenshots and more information about specific tests can be found in the section below correlation to the section number column of any specific test.

Section #	Description	Test	Expected Output	Success
1	The program runs without crashing.	Run the program.	The program does not crash upon starting.	Passed
2	The program can echo characters being typed to the screen.	Run the program and type on the keyboard.	The terminal prints characters being typed.	Passed
3	Typing a capital E will submit the message to be translated and then echoed to the screen.	Type a message and then type a capital ‘E’ character into the program.	The message before the ‘E’ is printed a second time.	Passed
4	After typing a capital E and sending the message to translate, all small ‘a’ characters will become ‘z’ characters in the translated output.	Type “aE” into the program.	The second line has a ‘z’ character and only that character.	Passed
5	After typing a bunch of small ‘a’ characters followed by a capital ‘E’, the entire line will be translated. This is to test multiple translated characters.	Type a message filled with multiple ‘a’ characters and end it with a capital ‘E’ character.	The second line will contain multiple ‘z’ characters.	Passed
6	Translating multiple ‘a’ characters does not in any way affect the surrounding characters.	Type a message with multiple ‘a’ characters (along with other characters) and then end it with a capital ‘E’.	The second line will contain a phrase with all ‘a’ characters replaced with ‘z’ characters.	Passed
7	Typing a capital ‘X’ will result in erasing the previous character typed once it is translated. The ‘X’ is effectively a backspace character.	Typing “12345XE” into the program.	The second line will be “1234”.	Passed
8	Typing multiple ‘X’ characters will backspace multiple preceding characters.	Typing “12345XXXXE” into the program.	The second line will have the character “1”.	Passed

9	The ‘X’ character will backspace the ‘a’ character that is to be translated. Essentially, the translation of the ‘a’ character will not be affected by the backspacing.	Type a phrase ending in “aXbc” into the program.	A phrase ending with “bc” will be outputted into the second line.	Passed
10	The user may translate multiple lines separately. After typing a message and translating it, the console resets and allows the user to type again, which they can then resend the message with another ‘E’ character.	Type “line1E”, then “line2E”, then “line3E”.	Translates and outputs all three messages separately.	Passed
11	A capital ‘K’ character will clear everything in that line before translating it.	Typing “delete this lineKE”.	The second line should be empty.	Passed
12	A capital ‘K’ will clear everything before it, however information typed after the ‘K’ will stay intact.	Typing “beforeKafterE”	The second line should only display the word “after”	Passed
13	A capital ‘K’ will only clear the line it is on and will not affect the lines before it in the terminal, nor will it affect any future lines. Printing a first line, K’ing the second line and then printed a third will only clear the 2nd line being translated.	Typing three different messages, with the second message containing a capital K.	The first and third message being translated will not be affected by the second messages clearing.	Passed.
14	A capital ‘T’ will translate the information on the line ‘T’ is on, and then will terminate the program naturally, ending all processes.	Type a message and end it with a capital ‘T’	The line will be translated, and then the program will exit. The ‘top’ terminal command will show no run away processes.	Passed
15	CTRL-K will immediately kill all processes without translating the message. All processes should be dead.	Type a message and end it by holding the control key and pressing the ‘k’ key.	The line will not be translated and the program will exit immediately. The ‘top’ terminal command will show no run away processes.	Passed

Contents

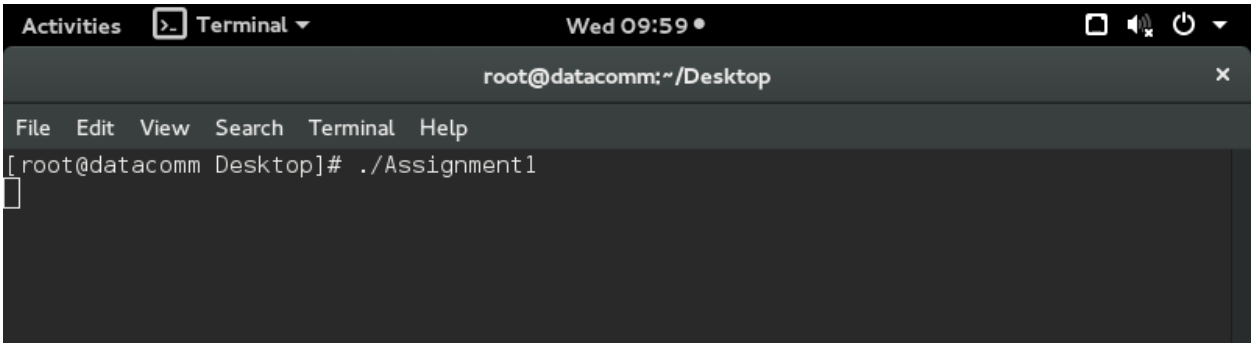
Test 1) Program Runs

Test Explanation: The program runs without crashing. We test this in the screenshot below by simply starting the program and seeing if it crashes.

Expected Output: The program does not crash upon starting.

Result: **Passed**

FIGURE 1: Program Output



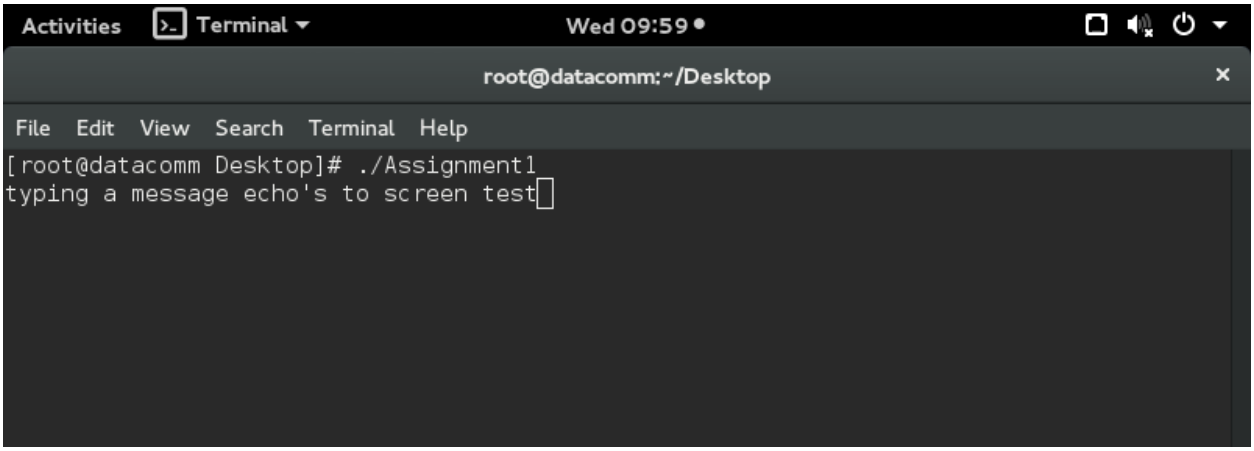
Test 2) Return Functionality to Echo

Test Explanation: The program can echo characters being typed to the screen. We test this in the screenshot below by running the program and type on the keyboard.

Expected Output: The terminal prints characters being typed.

Result: **Passed**

FIGURE 2: Program Output



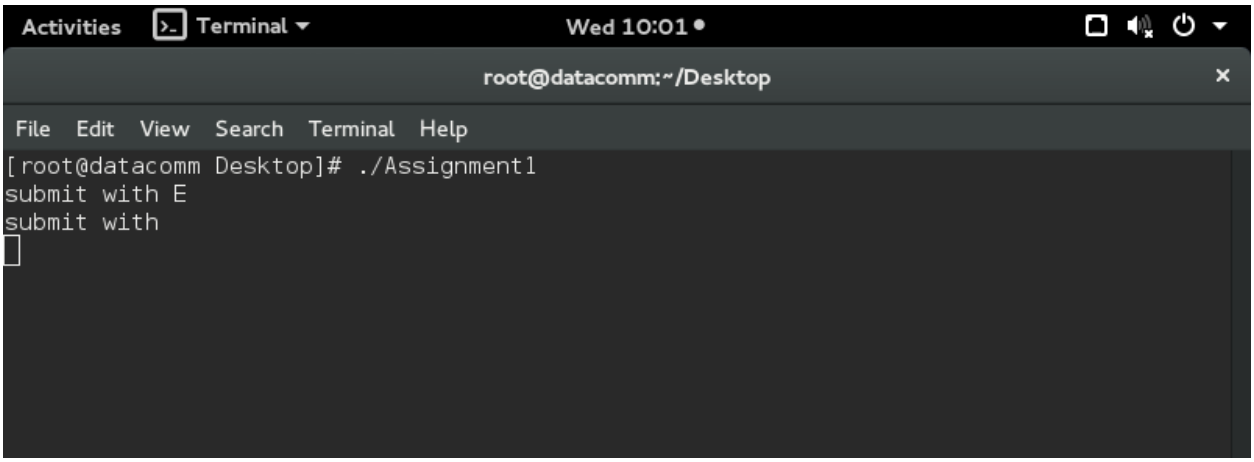
Test 3) Sends Data to be Translated

Test Explanation: Typing a capital E will submit the message to be translated and then echoed to the screen. We test this in the screenshot below by typing a message and then typing a capital ‘E’ character into the program.

Expected Output: The message before the ‘E’ is printed a second time.

Result: **Passed**

FIGURE 3: Program Output



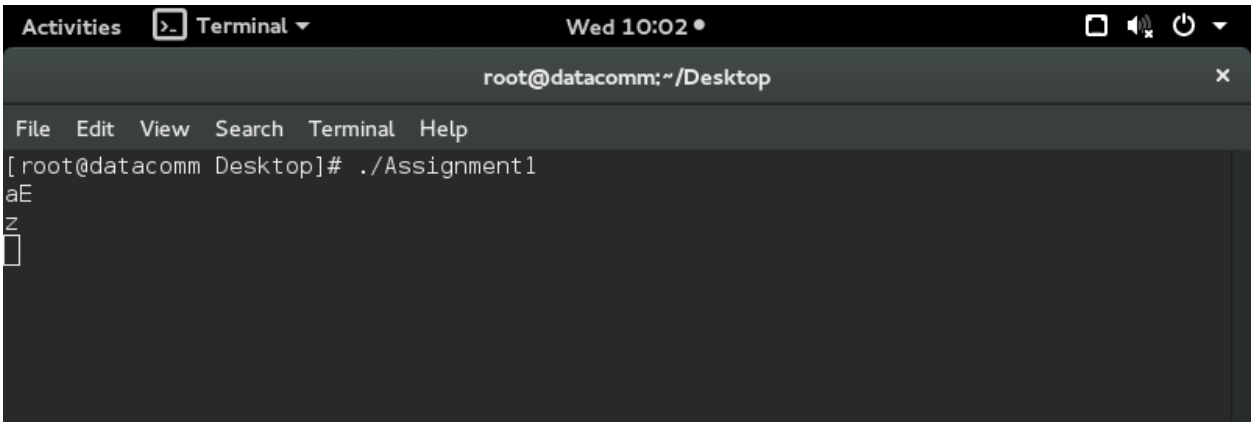
Test 4) Translates ‘a’ to ‘z’

Test Explanation: After typing a capital E and sending the message to translate, all small ‘a’ characters will become ‘z’ characters in the translated output. We test this in the screenshot below by typing “aE” into the program.

Expected Output: The second line has a ‘z’ character and only that character.

Result: **Passed**

FIGURE 4: Program Output



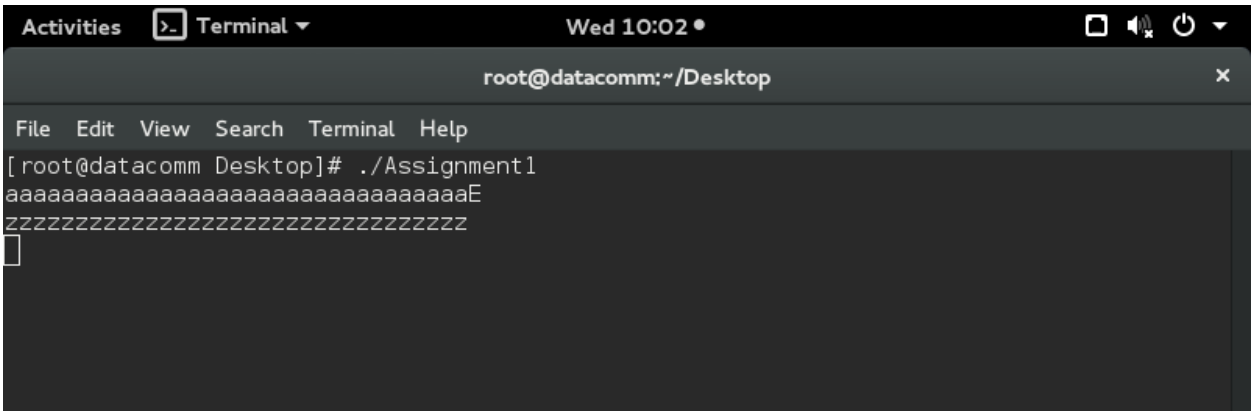
Test 5) Translates multiple ‘a’s to ‘z’s

Test Explanation: After typing a bunch of small ‘a’ characters followed by a capital ‘E’, the entire line will be translated. This is to test multiple translated characters. We test this in the screenshot below by typing a message filled with multiple ‘a’ characters and end it with a capital ‘E’ character.

Expected Output: The second line will contain multiple ‘z’ characters.

Result: **Passed**

FIGURE 5: Program Output



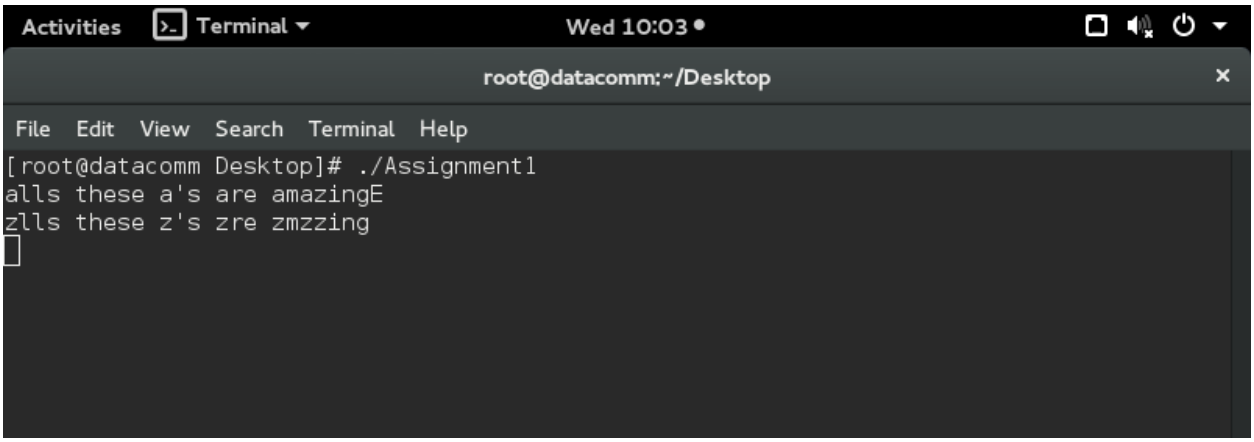
Test 6) Translations don’t affect non-translating characters.

Test Explanation: Translating multiple ‘a’ characters does not in any way affect the surrounding characters. We test this in the screenshot below by typing a message with multiple ‘a’ characters, along with other characters between them, and then end the message with a capital ‘E’.

Expected Output: The second line will contain a phrase with all ‘a’ characters replaced with ‘z’ characters.

Result: **Passed**

FIGURE 6: Program Output



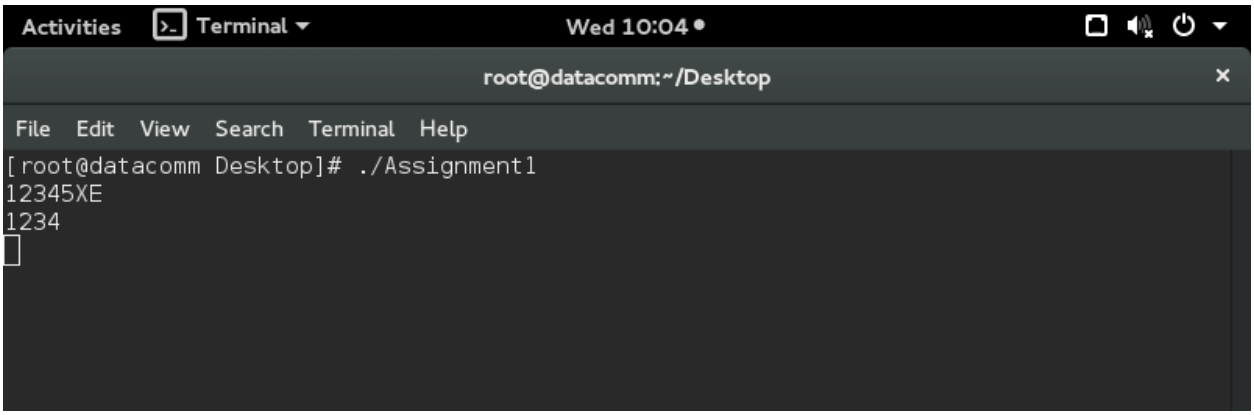
Test 7) ‘X’ is Treated as a Backspace

Test Explanation: Typing a capital ‘X’ will result in erasing the previous character typed once it is translated. The ‘X’ is effectively a backspace character. We test this in the screenshot below by typing “12345XE” into the program.

Expected Output: The second line will be “1234”.

Result: **Passed**

FIGURE 7: Program Output



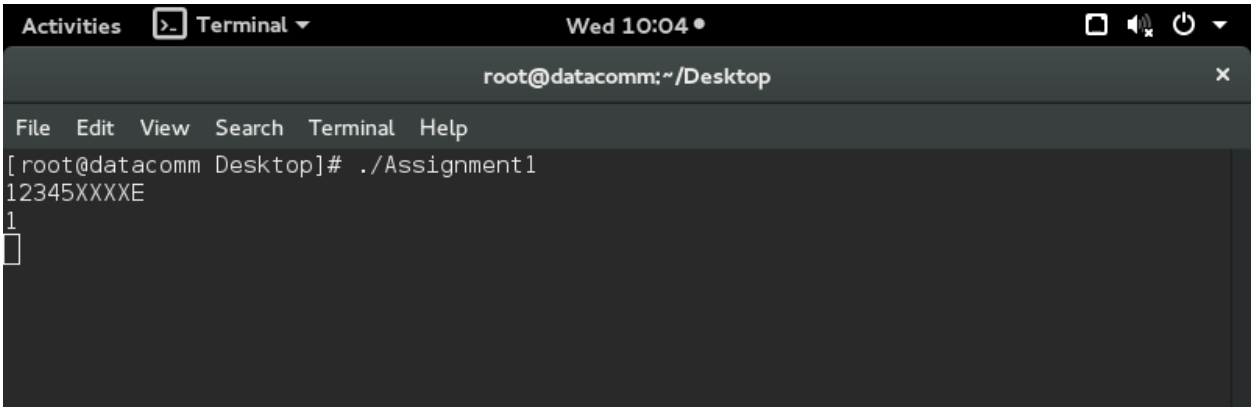
Test 8) Multiple ‘X’s Are Treated as Multiple Backspaces

Test Explanation: Typing multiple ‘X’ characters will backspace multiple preceding characters. We test this in the screenshot below by typing “12345XXXXE” into the program.

Expected Output: The second line will only show the number “1”.

Result: **Passed**

FIGURE 8: Program Output



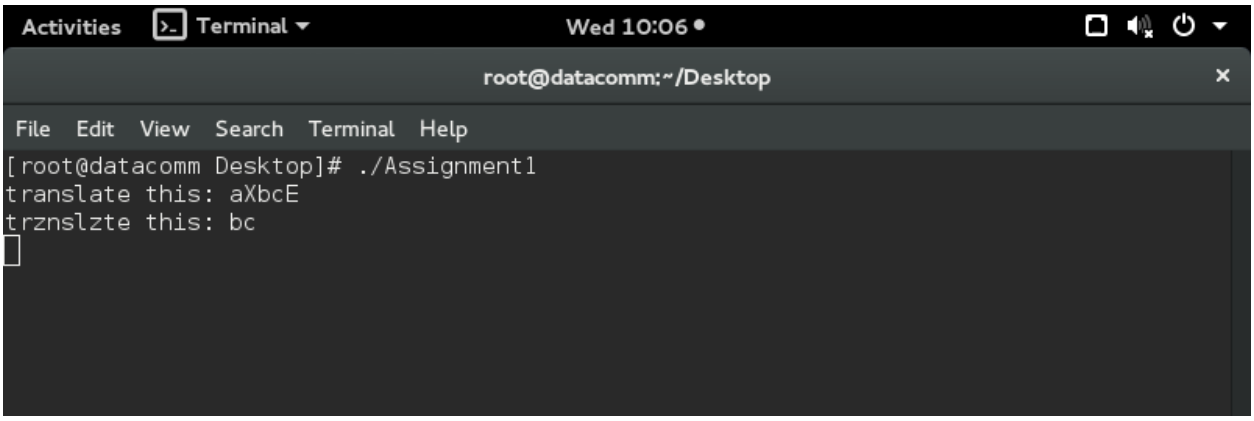
Test 9) Backspacing Overpowers Translating

Test Explanation: The 'X' character will backspace the 'a' character that is to be translated. Essentially, the translation of the 'a' character will not be affected by the backspacing. We test this in the screenshot below by typing a phrase ending in "aXbc" into the program.

Expected Output: The second line should be a translated phrase ending in "bc".

Result: **Passed**

FIGURE 9: Program Output



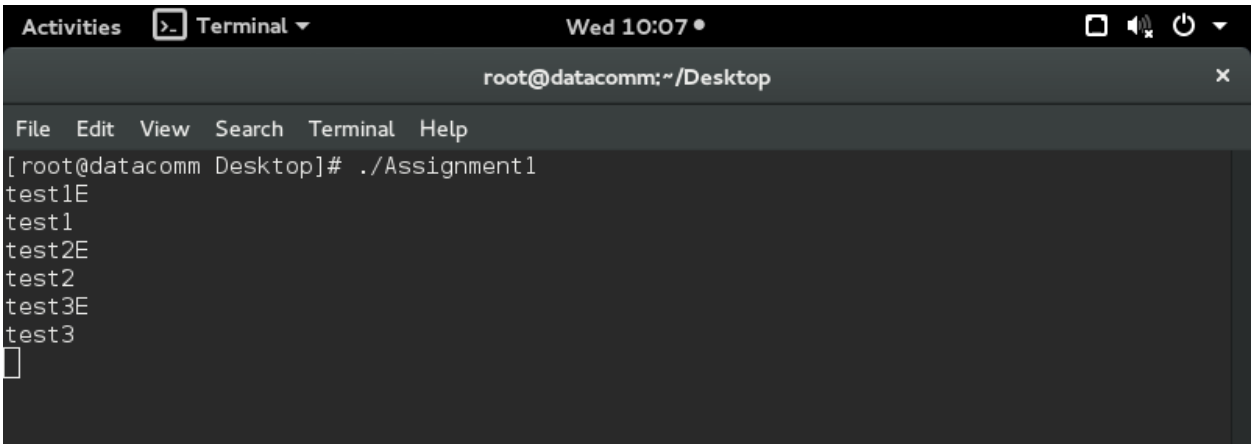
Test 10) Multiple Messages Can be Translated

Test Explanation: The user may translate multiple lines separately. After typing a message and translating it, the console resets and allows the user to type again, which they can then resend the message with another 'E' character. We test this in the screenshot below by typing "line1E", then "line2E", then "line3E".

Expected Output: Outputs three translated messages separately, one after the other.

Result: **Passed**

FIGURE 10: Program Output



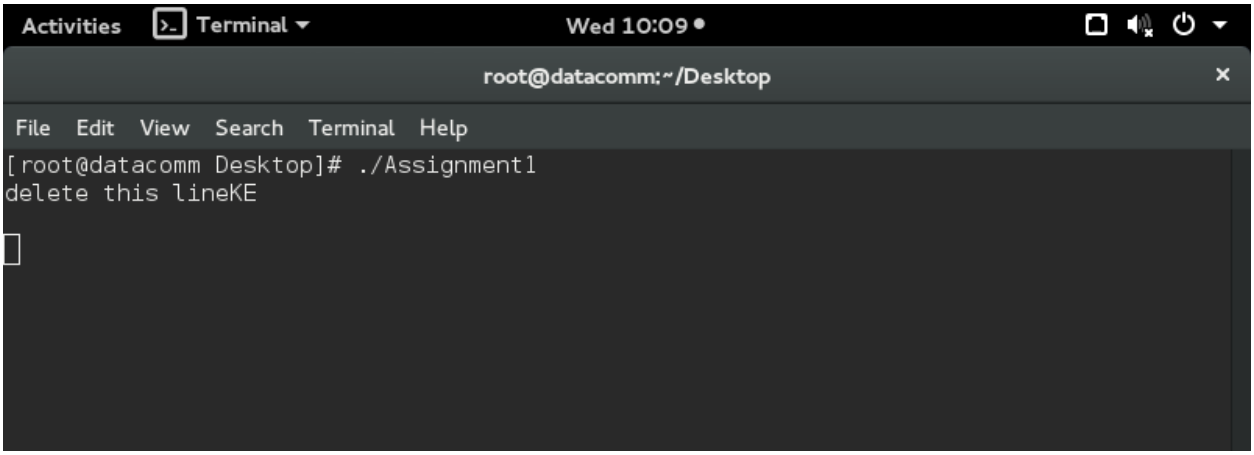
Test 11) Kill Line Clears the Line

Test Explanation: A capital 'K' character will clear everything in that line before translating it. We test this in the screenshot below by typing "delete this lineKE" into the program.

Expected Output: The second line should be an empty, blank line.

Result: **Passed**

FIGURE 11: Program Output



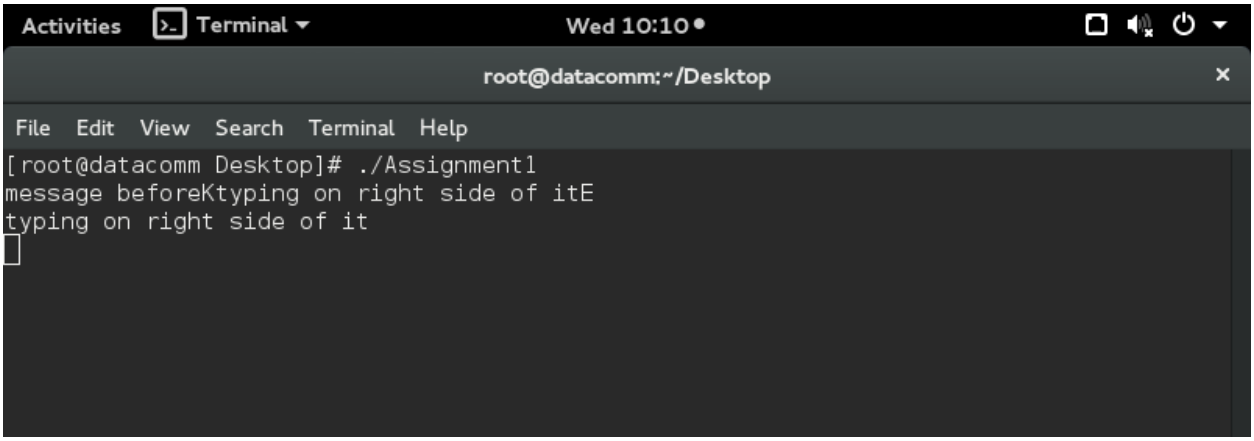
Test 12) Kill line Clears Only Before the ‘K’

Test Explanation: A capital ‘K’ will clear everything before it, however information typed after the ‘K’ will stay intact. We test this in the screenshot below by typing “beforeKafterE” into the program.

Expected Output: The second line should only display the word “after”.

Result: **Passed**

FIGURE 12: Program Output



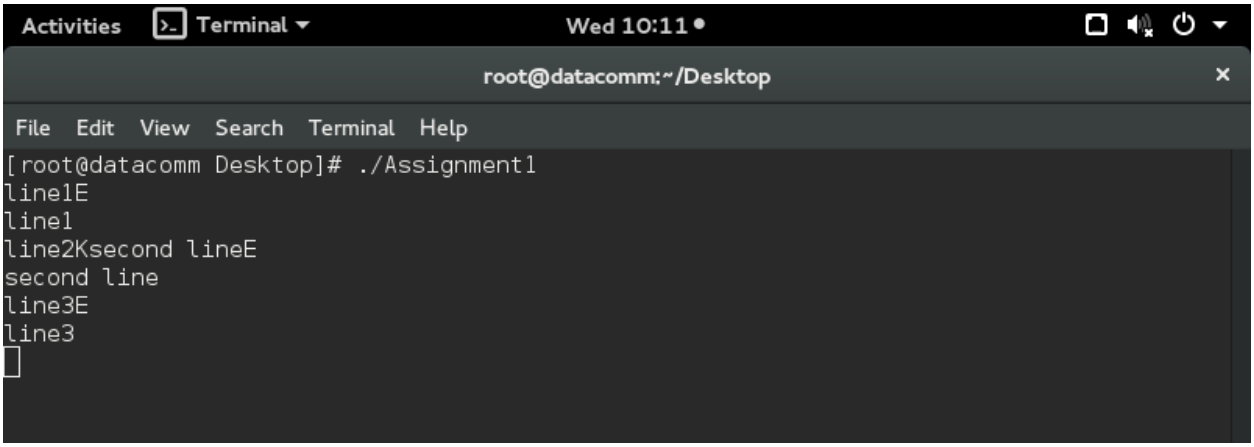
Test 13) Kill Line Does Not Affect Other Lines

Test Explanation: A capital 'K' will only clear the line it is on and will not affect the lines before it in the terminal, nor will it affect any future lines. Printing a first line, K'ing the second line and then printed a third will only clear the 2nd line being translated. We test this in the screenshot below by typing three different messages, with the second message containing a capital K.

Expected Output: The first and second messages translated will not be affected by the second message being cleared halfway through translation.

Result: **Passed**

FIGURE 13: Program Output



Test 14) Terminating the Program

Test Explanation: A capital ‘T’ will translate the information on the line ‘T’ is on, and then will terminate the program naturally, ending all processes. We test this in the screenshot be low by typing a message and ending that message with a capital ‘T’ character. We test that all processes have ended by running the terminals ‘top’ command.

Expected Output: The program will translate the message before the ‘T’ character, and then exit the program. It will successfully kill all processes.

Result: **Passed**

FIGURE 14a: Program Output

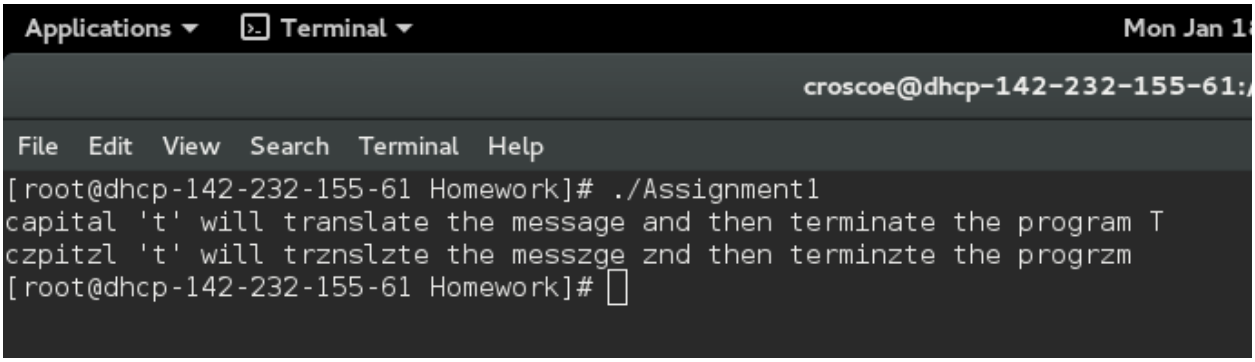
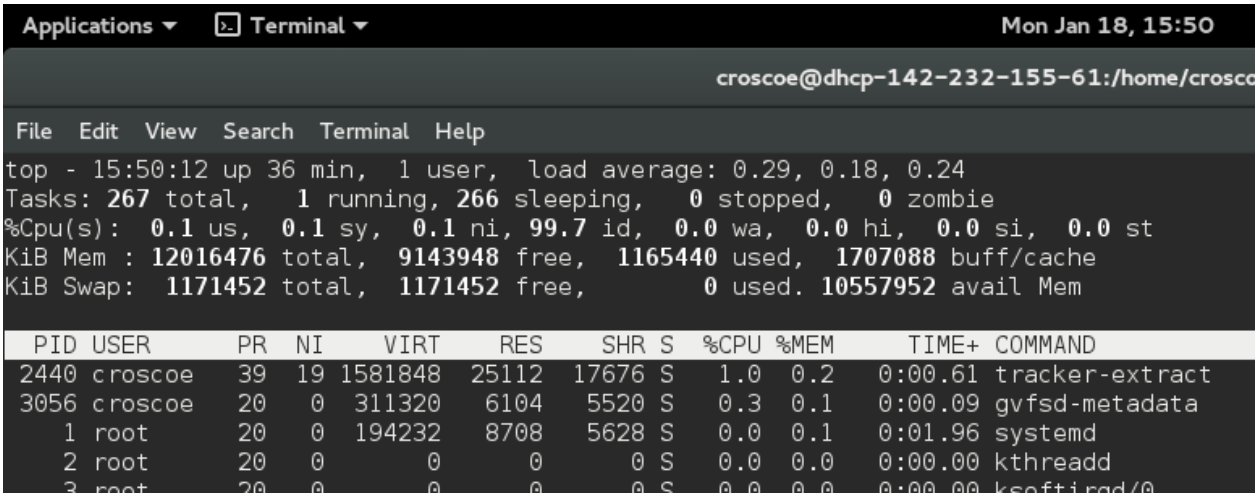


FIGURE 14b: Terminals ‘top’ command demonstrating that none of our processes are still alive doing work.



Test 15) Killing All Processes Immediately

Test Explanation: CTRL-K will immediately kill all processes without translating the message. All processes should be dead. We test this in the screenshots below by typing a message and end it by holding the control key and pressing the ‘k’ key. We test that the processes have been successfully killed via the terminals ‘top’ command.

Expected Output: The program will end immediately. All processes will have been killed.

Result: **Passed**

FIGURE 15a: Program Output

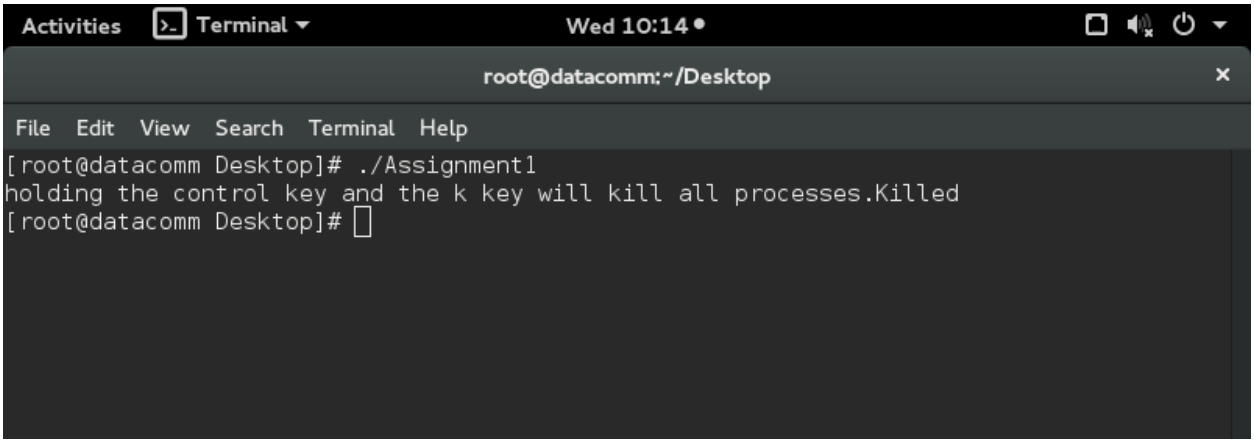


FIGURE 15b: Terminals ‘top’ command demonstrating that none of our processes are still alive doing work.