

# 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

<b>9</b>	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
<b>10</b>	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
<b>11</b>	A capital 'K' character will clear everything in that line before translating it.	Typing "delete this lineKE".	The second line should be empty.	Passed
<b>12</b>	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
<b>13</b>	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 2 <sup>nd</sup> 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.
<b>14</b>	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
<b>15</b>	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 .....	4
Test 2) Return Functionality to Echo.....	5
Test 3) Sends Data to be Translated .....	6
Test 4) Translates 'a' to 'z' .....	7
Test 5) Translates multiple 'a's to 'z's.....	8
Test 6) Translations don't affect non-translating characters. ....	9
Test 7) 'X' is Treated as a Backspace.....	10
Test 8) Multiple 'X's Are Treated as Multiple Backspaces .....	11
Test 9) Backspacing Overpowers Translating .....	12
Test 10) Multiple Messages Can be Translated .....	13
Test 11) Kill Line Clears the Line.....	14
Test 12) Kill line Clears Only Before the 'K' .....	15
Test 13) Kill Line Does Not Affect Other Lines .....	16
Test 14) Terminating the Program.....	17
Test 15) Killing All Processes Immediately.....	18

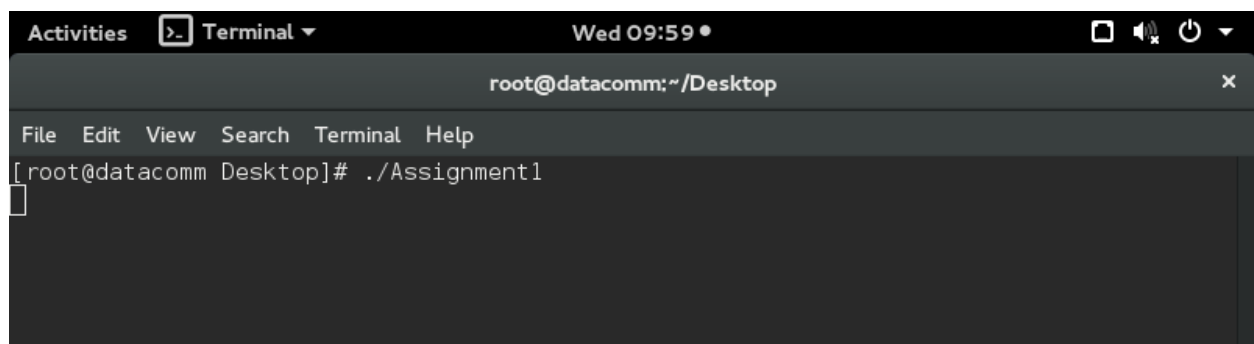
## Test 1) Program Runs

**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



The screenshot shows a terminal window titled "Terminal" with a subtitle "root@datacomm: ~/Desktop". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal prompt is "[root@datacomm Desktop]# ./Assignment1" and a cursor is visible on the line below.

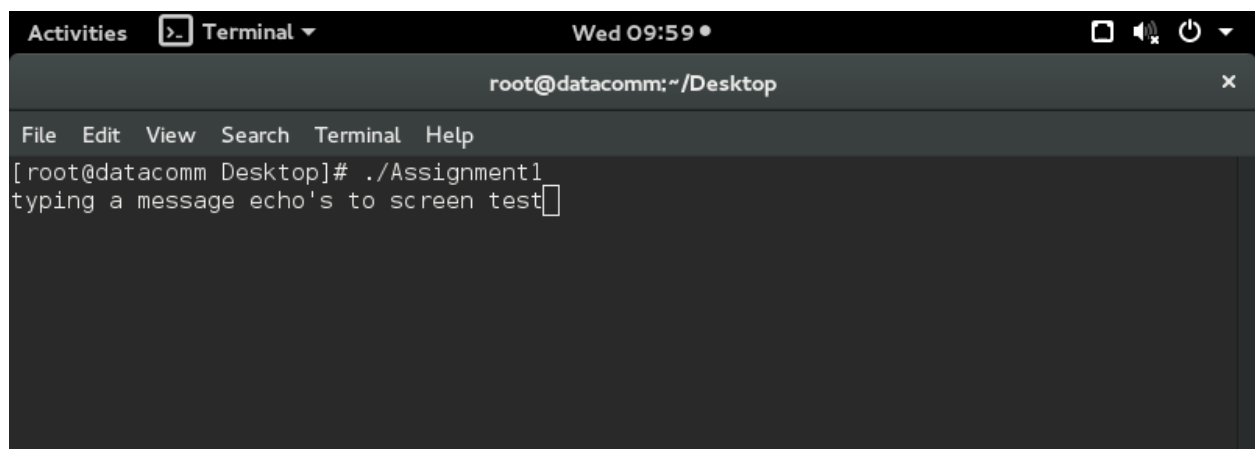
## Test 2) Return Functionality to Echo

**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

A screenshot of a Linux terminal window. The title bar at the top shows 'Activities', a terminal icon, 'Terminal', and the date 'Wed 09:59'. Below the title bar, the terminal's address bar shows 'root@datacomm: ~/Desktop'. The terminal has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main area of the terminal shows the command prompt '[root@datacomm Desktop]# ./Assignment1' followed by the text 'typing a message echo's to screen test' and a cursor. The text is displayed in a light gray font on a dark background.

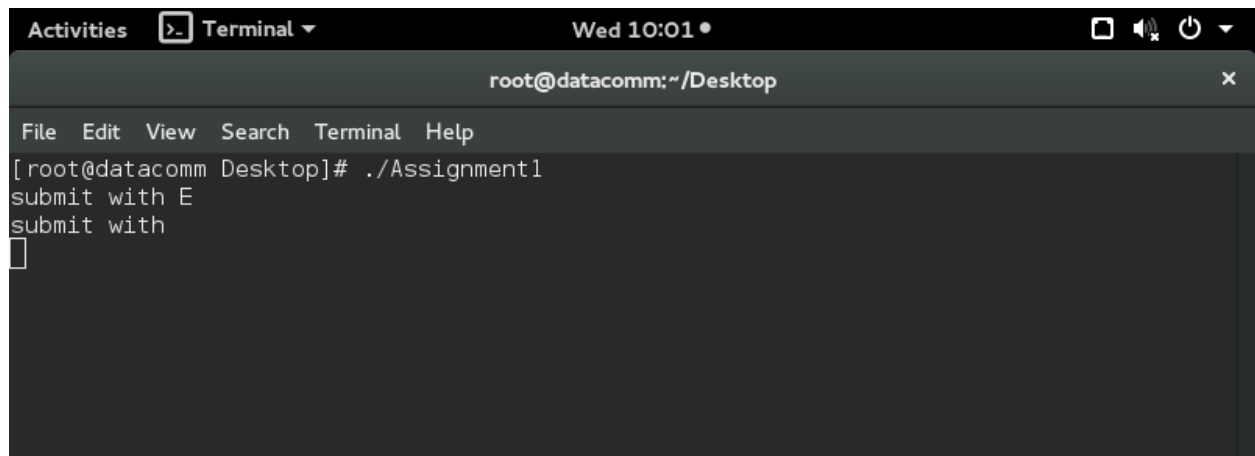
## Test 3) Sends Data to be Translated

**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

A screenshot of a terminal window. The title bar shows 'Activities', 'Terminal', and 'Wed 10:01'. The terminal content shows the prompt '[root@datacomm Desktop]# ./Assignment1' followed by the output 'submit with E' and 'submit with' on two separate lines. A cursor is visible at the end of the second line.

```
Activities Terminal Wed 10:01
root@datacomm:~/Desktop
File Edit View Search Terminal Help
[root@datacomm Desktop]# ./Assignment1
submit with E
submit with
█
```

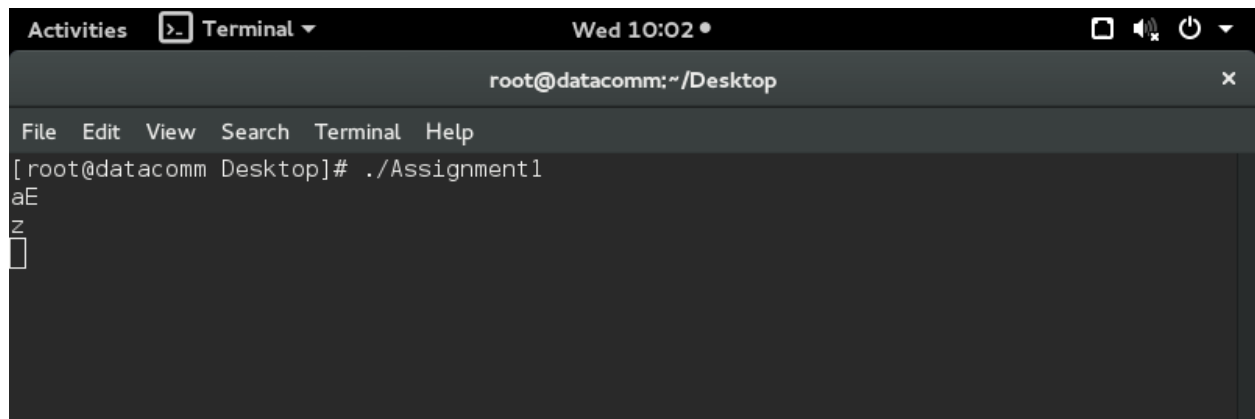
## Test 4) Translates 'a' to 'z'

**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



```
Activities Terminal Wed 10:02
root@datacomm: ~/Desktop
File Edit View Search Terminal Help
[root@datacomm Desktop]# ./Assignment1
aE
z

```

### 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

A screenshot of a Linux terminal window. The title bar at the top shows "Activities", a terminal icon, "Terminal", and system status icons (network, volume, power) along with the time "Wed 10:02". Below the title bar, the terminal's working directory is shown as "root@datacomm: ~/Desktop". A menu bar contains "File", "Edit", "View", "Search", "Terminal", and "Help". The main area displays the command prompt "[root@datacomm Desktop]# ./Assignment1", followed by two lines of output: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaE" and "zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz". A cursor is visible on the line following the second output line.



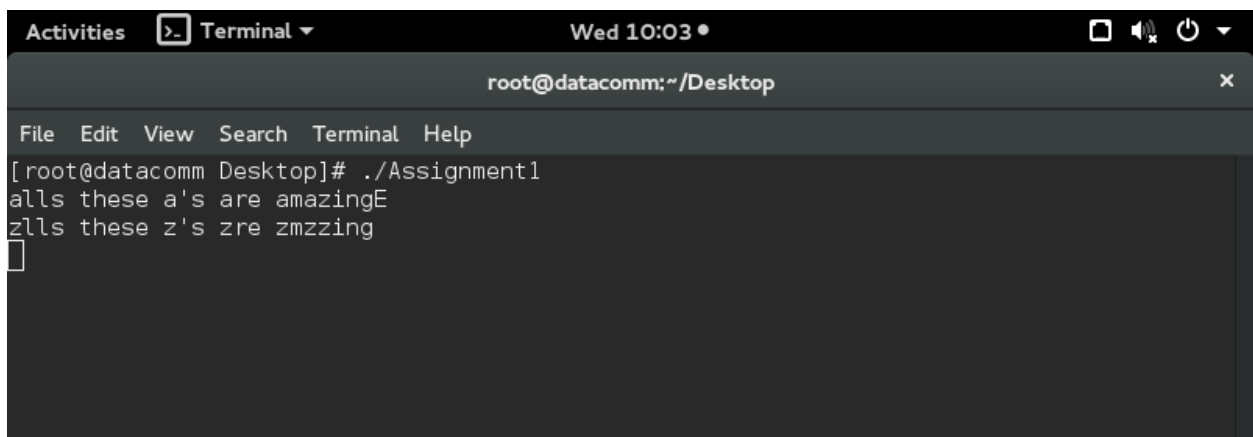
## Test 6) Translations don't affect non-translating characters.

**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



The screenshot shows a terminal window titled "Terminal" with a subtitle "root@datacomm: ~/Desktop". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows the following commands and output:

```
[root@datacomm Desktop]# ./Assignment1
alls these a's are amazingE
zlls these z's zre zmzzing
```

The output consists of two lines. The first line is "alls these a's are amazingE" and the second line is "zlls these z's zre zmzzing". The cursor is positioned at the end of the second line.

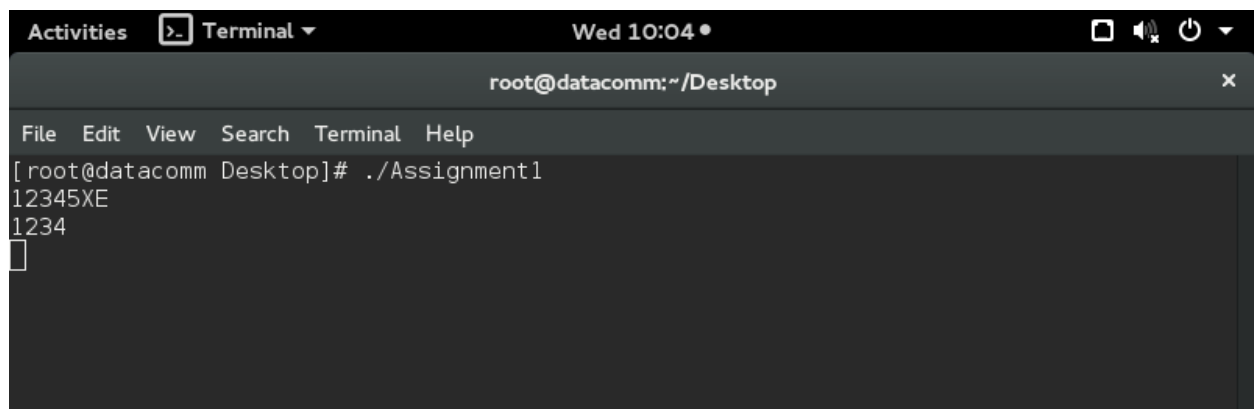
## Test 7) 'X' is Treated as a Backspace

**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



The screenshot shows a terminal window titled "Terminal" with a subtitle "root@datacomm: ~/Desktop". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows the command `./Assignment1` being executed, followed by the output `12345XE` on the first line and `1234` on the second line. A cursor is visible at the end of the second line.

```
[root@datacomm Desktop]# ./Assignment1
12345XE
1234
█
```

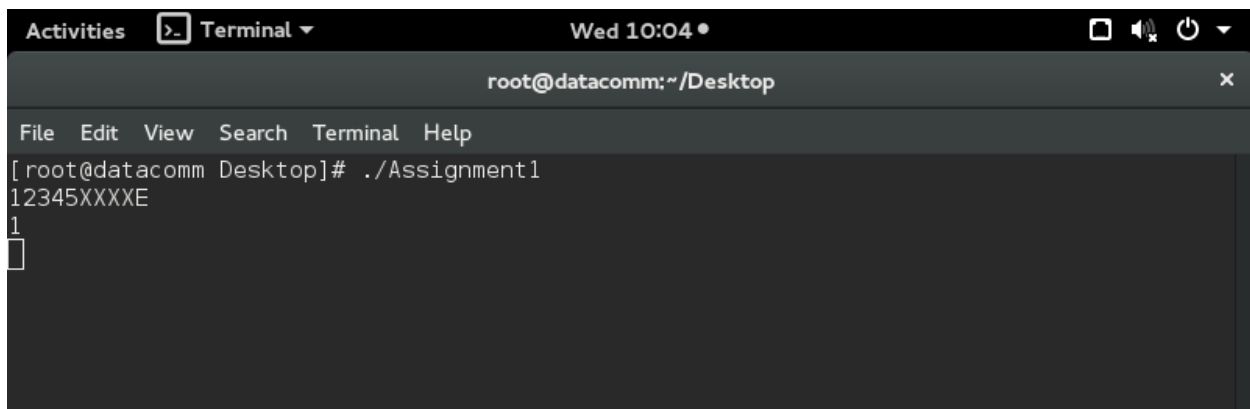
## Test 8) Multiple 'X's Are Treated as Multiple Backspaces

**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



```
Activities  Terminal  Wed 10:04
root@datacomm:~/Desktop
File Edit View Search Terminal Help
[root@datacomm Desktop]# ./Assignment1
12345XXXXE
1
█
```

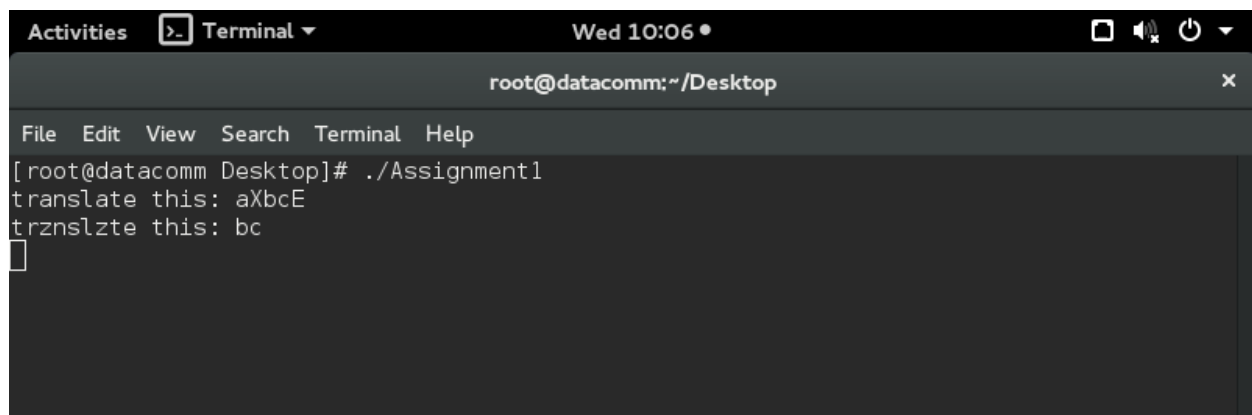
## Test 9) Backspacing Overpowers Translating

**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



The screenshot shows a terminal window titled "Terminal" with a subtitle "root@datacomm: ~/Desktop". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows the following commands and output:

```
[root@datacomm Desktop]# ./Assignment1
translate this: aXbcE
trznslzte this: bc
█
```

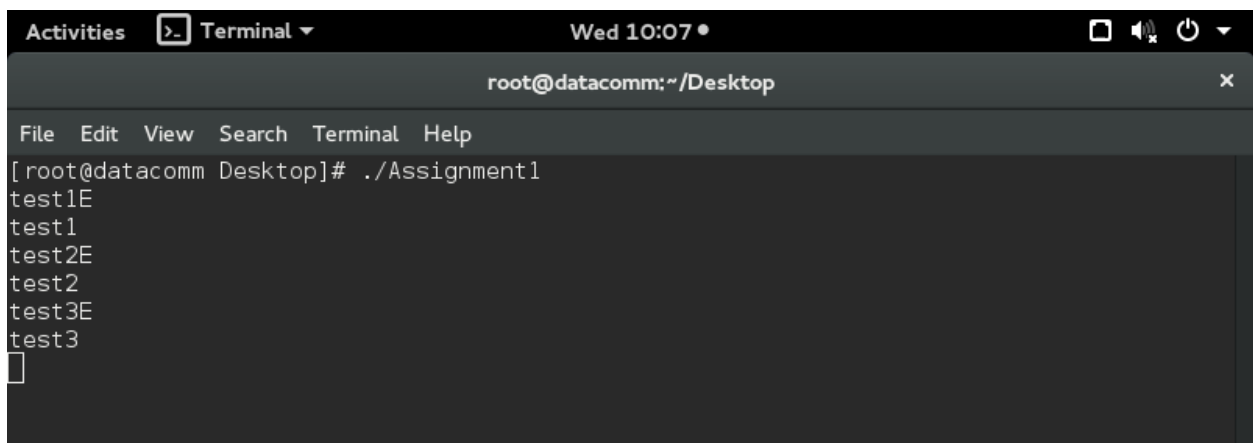
## Test 10) Multiple Messages Can be Translated

**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



The screenshot shows a terminal window titled "Terminal" with a timestamp of "Wed 10:07". The terminal prompt is "root@datacomm: ~/Desktop". The user has entered the command " ./Assignment1". The output of the program is as follows:

```
[root@datacomm Desktop]# ./Assignment1
test1E
test1
test2E
test2
test3E
test3

```

The terminal shows the program output for Test 10. The user enters "test1E", and the program outputs "test1". Then the user enters "test2E", and the program outputs "test2". Finally, the user enters "test3E", and the program outputs "test3". The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal title bar shows "root@datacomm: ~/Desktop".

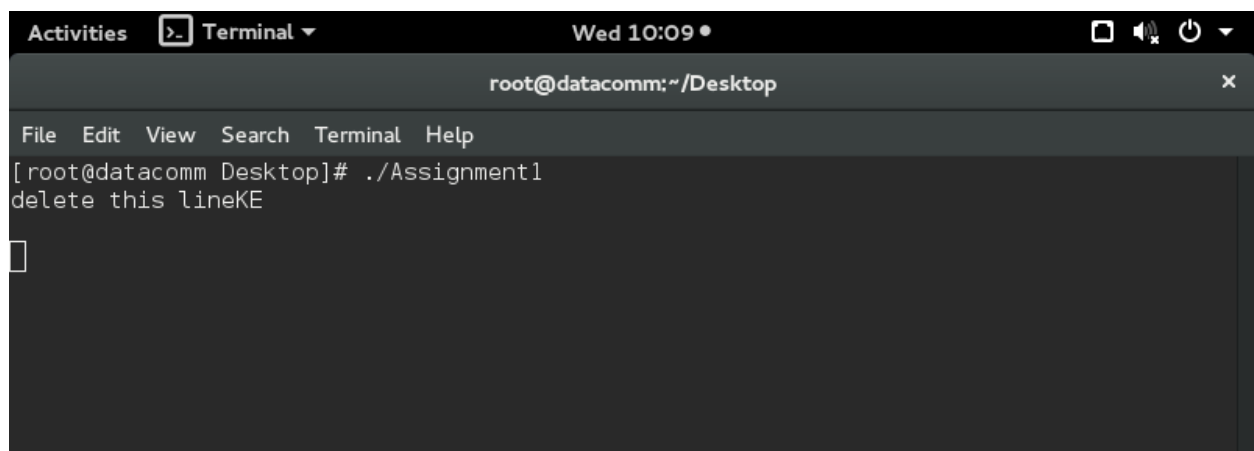
## Test 11) Kill Line Clears the Line

**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



```
Activities  Terminal  Wed 10:09
root@datacomm:~/Desktop
File Edit View Search Terminal Help
[root@datacomm Desktop]# ./Assignment1
delete this lineKE

```

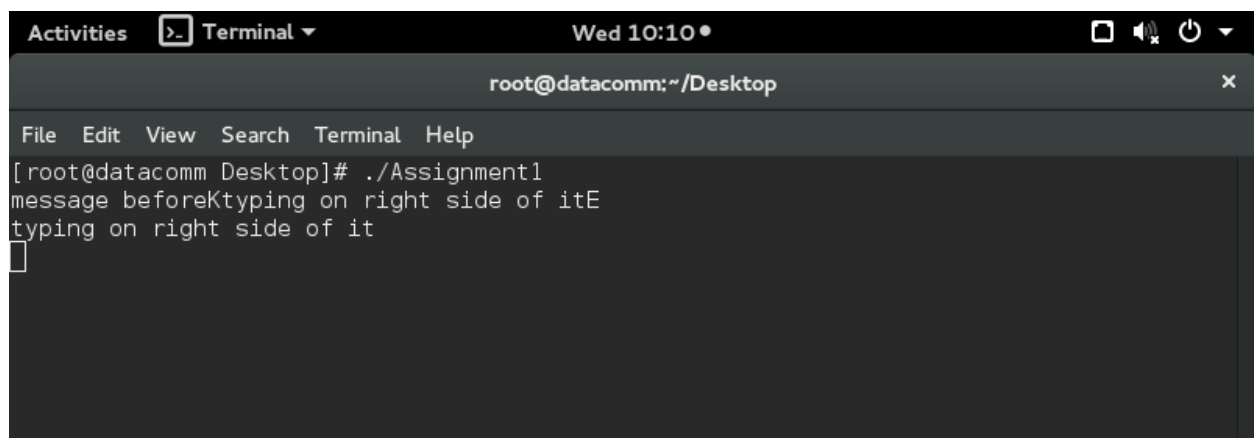
## Test 12) Kill line Clears Only Before the 'K'

**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



The screenshot shows a terminal window titled "Terminal" with a timestamp of "Wed 10:10". The terminal prompt is "root@datacomm: ~/Desktop". The output of the program is as follows:

```
[root@datacomm Desktop]# ./Assignment1
message beforeKtyping on right side of itE
typing on right side of it
█
```

The output consists of two lines. The first line is "message beforeKtyping on right side of itE". The second line is "typing on right side of it". The cursor is positioned at the end of the second line, indicated by a small white square.

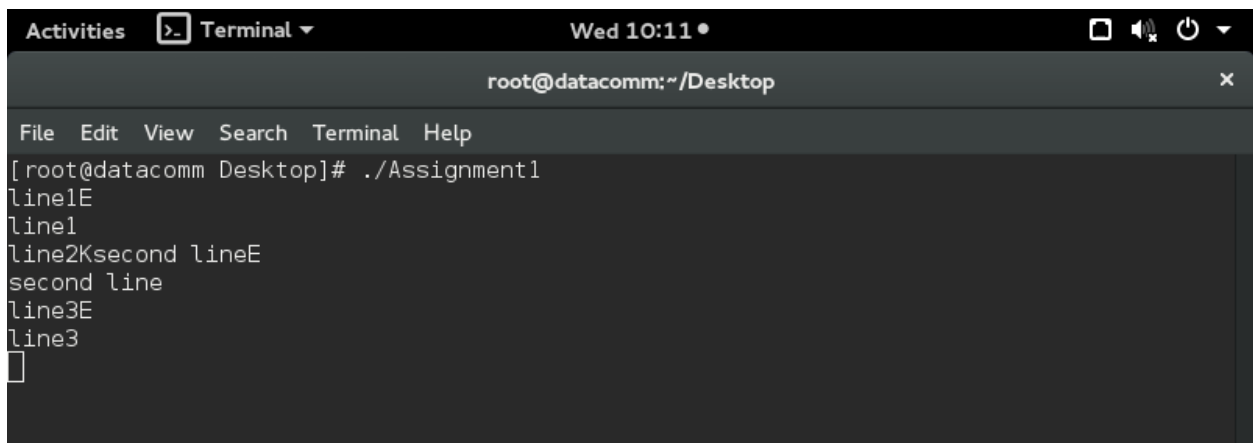
## Test 13) Kill Line Does Not Affect Other Lines

**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 2<sup>nd</sup> 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



The screenshot shows a terminal window titled "Terminal" with the current directory set to "~/Desktop". The prompt is "root@datacomm:". The user has executed the command "./Assignment1". The output consists of several lines: "line1E", "line1", "line2Ksecond lineE", "second line", "line3E", and "line3". A cursor is visible at the end of the last line.

```
root@datacomm:~/Desktop
File Edit View Search Terminal Help
[root@datacomm Desktop]# ./Assignment1
line1E
line1
line2Ksecond lineE
second line
line3E
line3
█
```



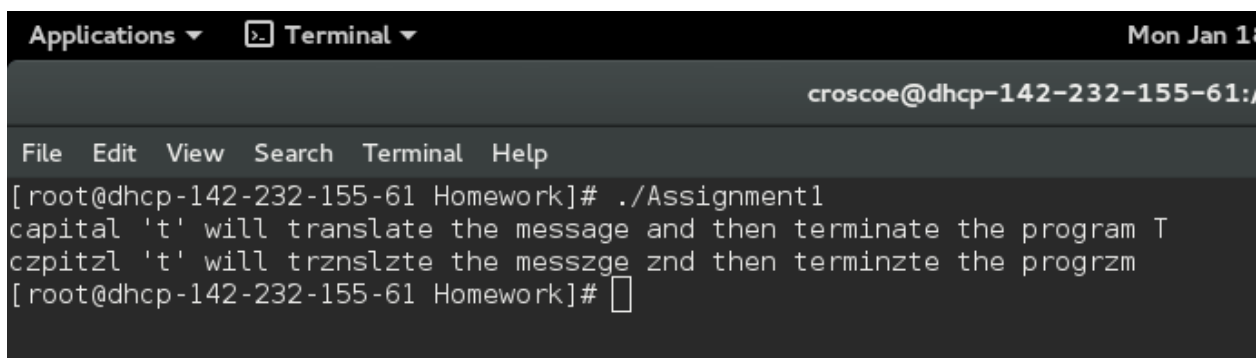
## Test 14) Terminating the Program

**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 below by typing a message and ending that message with a capital 'T' character. We test that all processes have ended by running the terminal's '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.

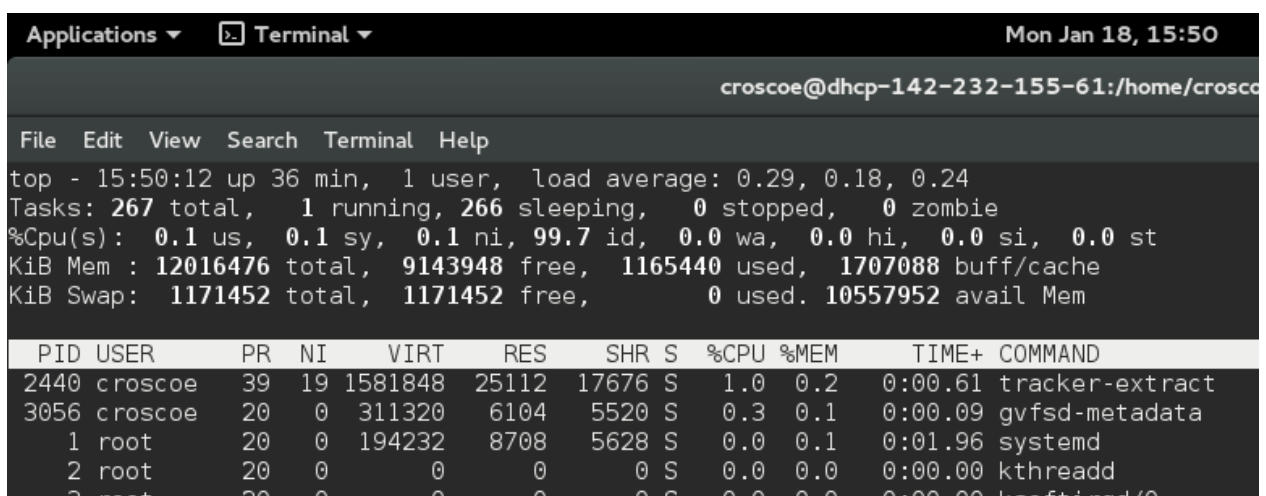
**Result: Passed**

**FIGURE 14a:** Program Output

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar (Applications, Terminal, Mon Jan 18, 15:50). The prompt is 'croscoe@dhcp-142-232-155-61:'. The user has entered './Assignment1' and the program has output two lines of text: 'capital 't' will translate the message and then terminate the program T' and 'czpitzl 't' will trznslzte the messzge znd then terminzte the progrzm'. The prompt is now '[root@dhcp-142-232-155-61 Homework]# ' followed by a cursor.

```
Applications ▾ Terminal ▾ Mon Jan 18, 15:50
croscoe@dhcp-142-232-155-61:
File Edit View Search Terminal Help
[root@dhcp-142-232-155-61 Homework]# ./Assignment1
capital 't' will translate the message and then terminate the program T
czpitzl 't' will trznslzte the messzge znd then terminzte the progrzm
[root@dhcp-142-232-155-61 Homework]#
```

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

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar (Applications, Terminal, Mon Jan 18, 15:50). The prompt is 'croscoe@dhcp-142-232-155-61:/home/croscoe'. The user has entered 'top' and the command has output system statistics and a table of running processes. The statistics show 1 user, 1 running process, and 266 sleeping processes. The table lists processes with columns: PID, USER, PR, NI, VIRT, RES, SHR, S, %CPU, %MEM, TIME+, and COMMAND.

```
Applications ▾ Terminal ▾ Mon Jan 18, 15:50
croscoe@dhcp-142-232-155-61:/home/croscoe
File Edit View Search Terminal Help
top - 15:50:12 up 36 min, 1 user, load average: 0.29, 0.18, 0.24
Tasks: 267 total, 1 running, 266 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.1 us, 0.1 sy, 0.1 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 12016476 total, 9143948 free, 1165440 used, 1707088 buff/cache
KiB Swap: 1171452 total, 1171452 free, 0 used. 10557952 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 2440 croscoe   39   19 1581848 25112 17676 S   1.0   0.2   0:00.61 tracker-extract
 3056 croscoe   20    0 311320   6104 5520 S   0.3   0.1   0:00.09 gvfsd-metadata
    1 root      20    0 194232   8708 5628 S   0.0   0.1   0:01.96 systemd
    2 root      20    0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root      20    0      0      0      0 S   0.0   0.0   0:00.00 ksoftirqd/0
```

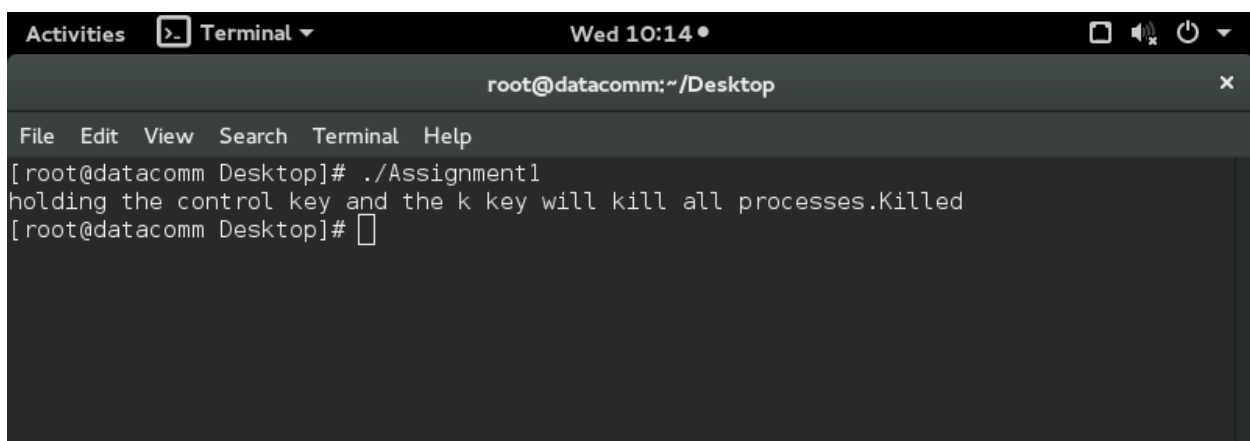
## Test 15) Killing All Processes Immediately

**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.

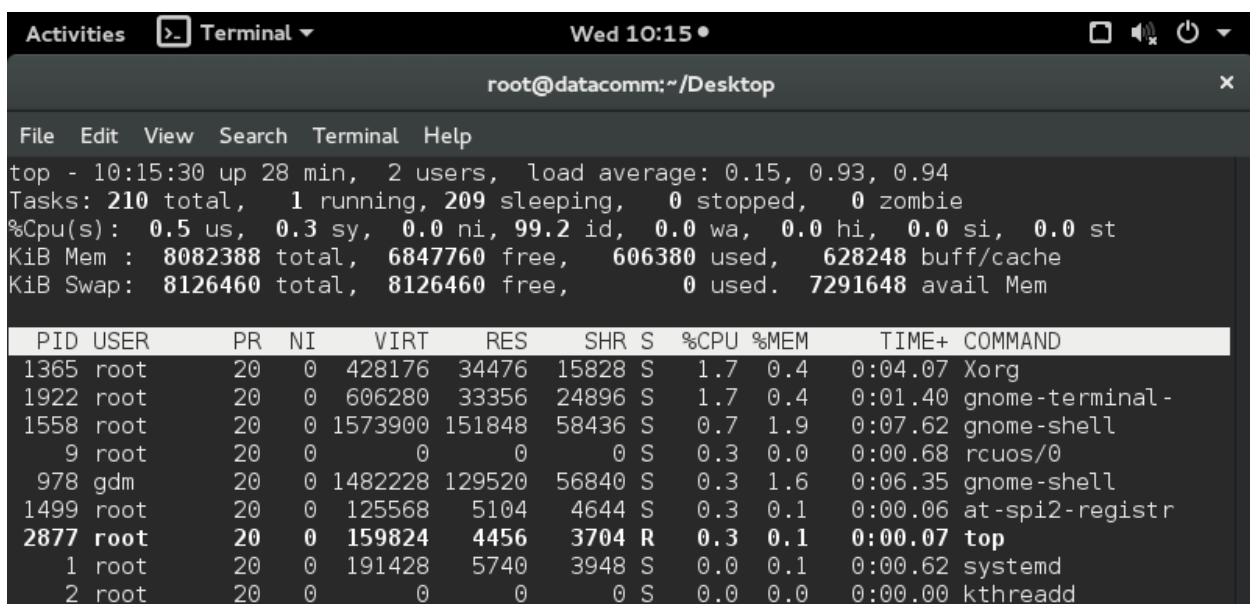
**Result:** Passed

**FIGURE 15a:** Program Output

A terminal window titled 'root@datacomm: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command './Assignment1' being executed, followed by the message 'holding the control key and the k key will kill all processes.Killed' and a prompt for the next command.

```
root@datacomm:~/Desktop
File Edit View Search Terminal Help
[root@datacomm Desktop]# ./Assignment1
holding the control key and the k key will kill all processes.Killed
[root@datacomm Desktop]#
```

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

A terminal window titled 'root@datacomm: ~/Desktop' showing the output of the 'top' command. The output displays system statistics and a list of running processes. The 'top' process itself is highlighted in bold.

```
root@datacomm:~/Desktop
File Edit View Search Terminal Help
top - 10:15:30 up 28 min, 2 users, load average: 0.15, 0.93, 0.94
Tasks: 210 total, 1 running, 209 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.5 us, 0.3 sy, 0.0 ni, 99.2 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8082388 total, 6847760 free, 606380 used, 628248 buff/cache
KiB Swap: 8126460 total, 8126460 free, 0 used, 7291648 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 1365 root        20   0  428176  34476 15828 S   1.7   0.4   0:04.07 Xorg
 1922 root        20   0  606280  33356 24896 S   1.7   0.4   0:01.40 gnome-terminal-
 1558 root        20   0 1573900 151848 58436 S   0.7   1.9   0:07.62 gnome-shell
    9 root        20   0     0     0     0 S   0.3   0.0   0:00.68 rcuos/0
   978 gdm         20   0 1482228 129520 56840 S   0.3   1.6   0:06.35 gnome-shell
 1499 root        20   0 1255568  5104  4644 S   0.3   0.1   0:00.06 at-spi2-registr
 2877 root        20   0 159824  4456  3704 R   0.3   0.1   0:00.07 top
    1 root        20   0 191428  5740  3948 S   0.0   0.1   0:00.62 systemd
    2 root        20   0     0     0     0 S   0.0   0.0   0:00.00 kthreadd
```