<u>Pseudocode</u>

Input Process

```
Turn off all terminal processes
Catch terminal signals for SGABRT and SIGTERM to be handled
Turn off all terminal processes
Create pipe for Input / Output processes
Create the translate process by calling fork()
Check if result == 0
     Transition to translate state
Else
     Create the output process by calling fork()
     Check if result == 0
          Transition to output state
Else
     Transition to get input state
```

Get Input

```
Close the translator read descriptor
Close the output read descriptor
Loop forever
     Reads input from keyboard
     When inputs are read
          Transition to Read state
```

Read

```
Have a buffer that stores the current line
Write each character to the input / output pipe
     Check if abnormal termination character is read (^k)
          Sends the SIGABRT signal to all three processes
          Transition to End state
     Check if the carriage return character is read (E)
          Write the buffer to the input / translate pipe
          Re-initialize the buffer
     Else
          Store each character into the buffer
```

Output Process

```
Close the output write descriptor
Forever loop
     Keeps on reading message from the input / output pipe
     When something is read
           Transition to Display state
```

Display

```
Output string onto the terminal screen
```

Translate Process

```
Close the input / output read descriptor
Close the input / translate write descriptor
Forever loop
     Keeps on reading message from the input / translate pipe
     When something is read
           Transition to Translate state
```

Translate

```
Close the output read descriptor
Close the translate write descriptor
Have a buffer for message read from the input / translate pipe
Have a buffer for translated message for input / translated pipe
Loops through the message read from input /translate pipe
     If 'a' is read
           Replace it with 'z' and add it onto the translated buffer
     If backspace('X') is read
           Delete the previous character
     If line kill('K') is read
           Delete all preceding characters
     If the normal termination('T') is read
           Send SIGTERM to all three processes
     Else
           Adds each character to the translated buffer
Write the translated buffer to the input / output buffer
```