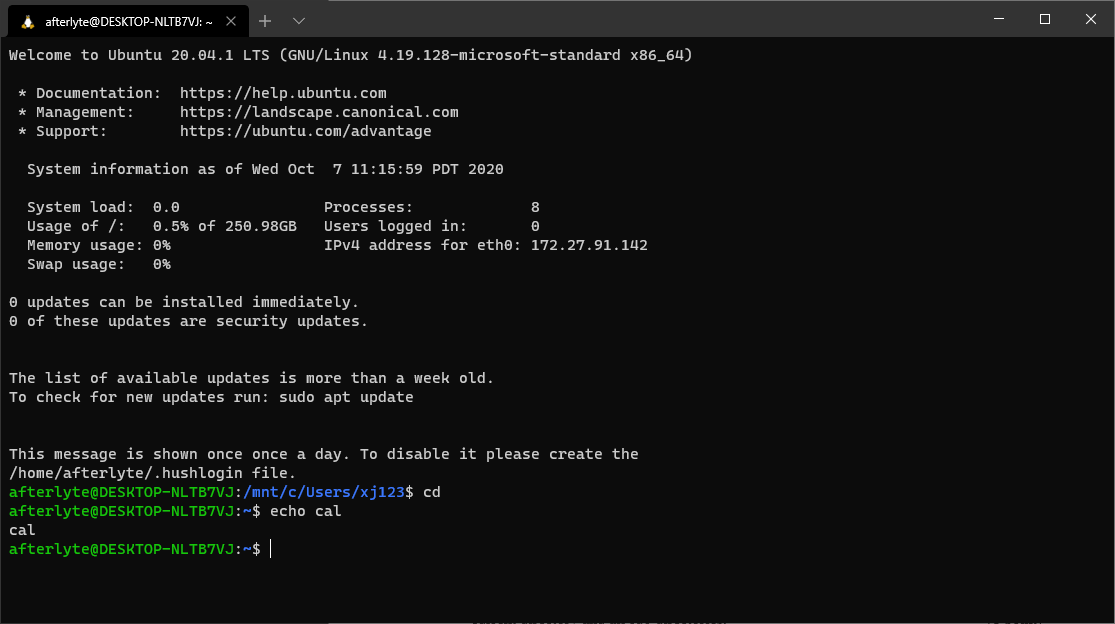
**Lab 2: More Unix Commands**

# ‘echo’

1. What are the differences among the following commands? Explain with screenshots.

(2 pts)

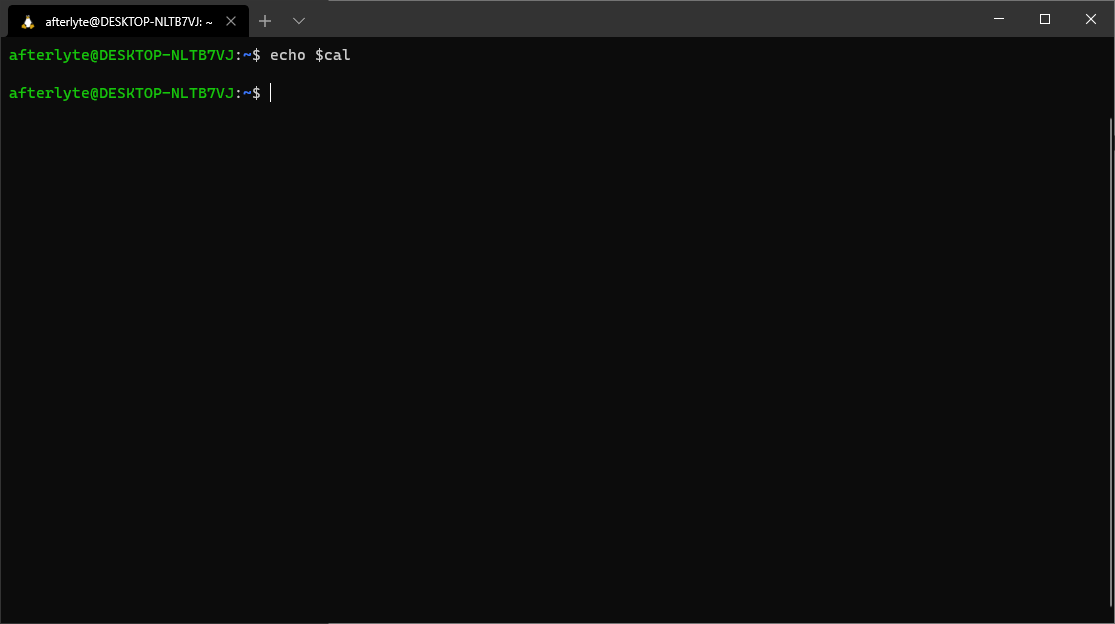
a) echo cal – prints the literal word “cal”



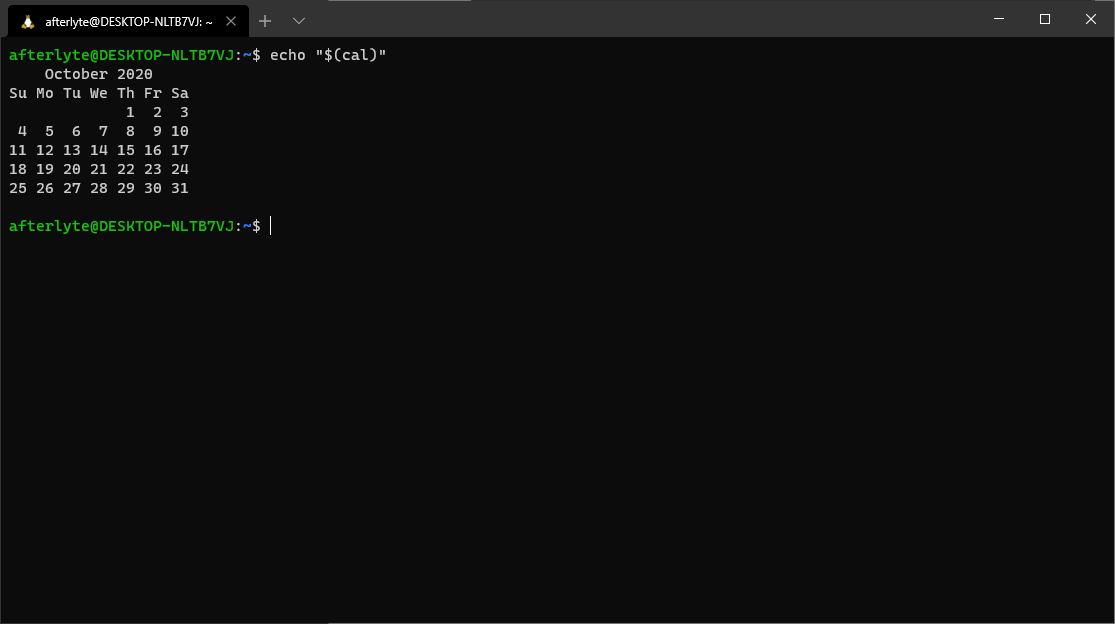
1. echo $(cal) – prints the result of the command ‘cal’ on one line



1. echo $cal – attempts to print the value of the variable ‘cal’ (currently there is none)



1. echo “$(cal)” – prints the results of the command ‘cal’, but formats it the way the original command is

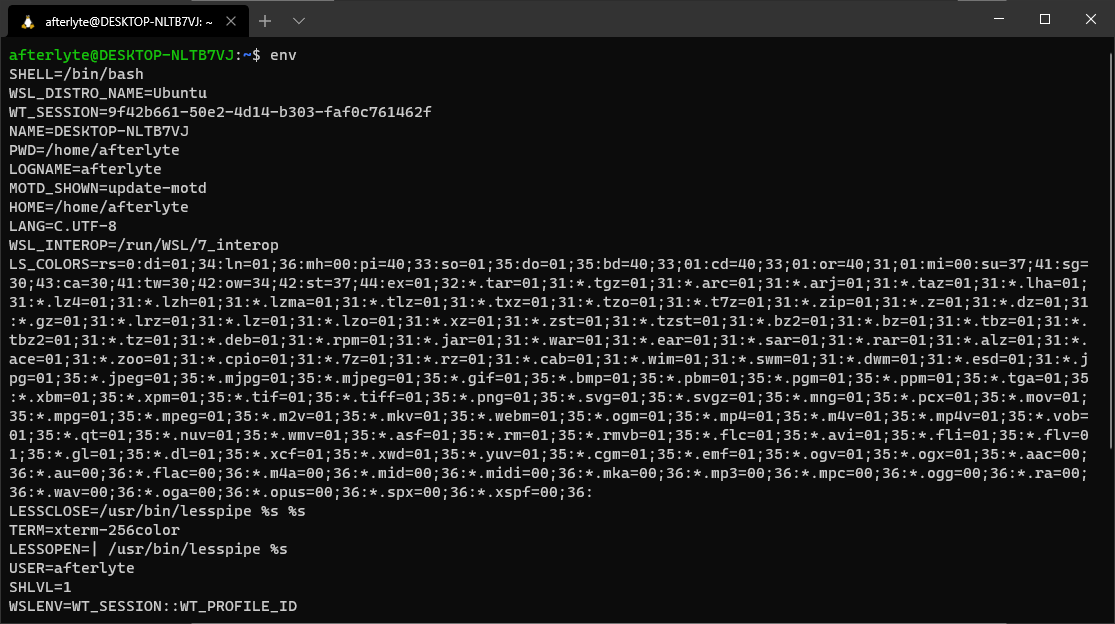


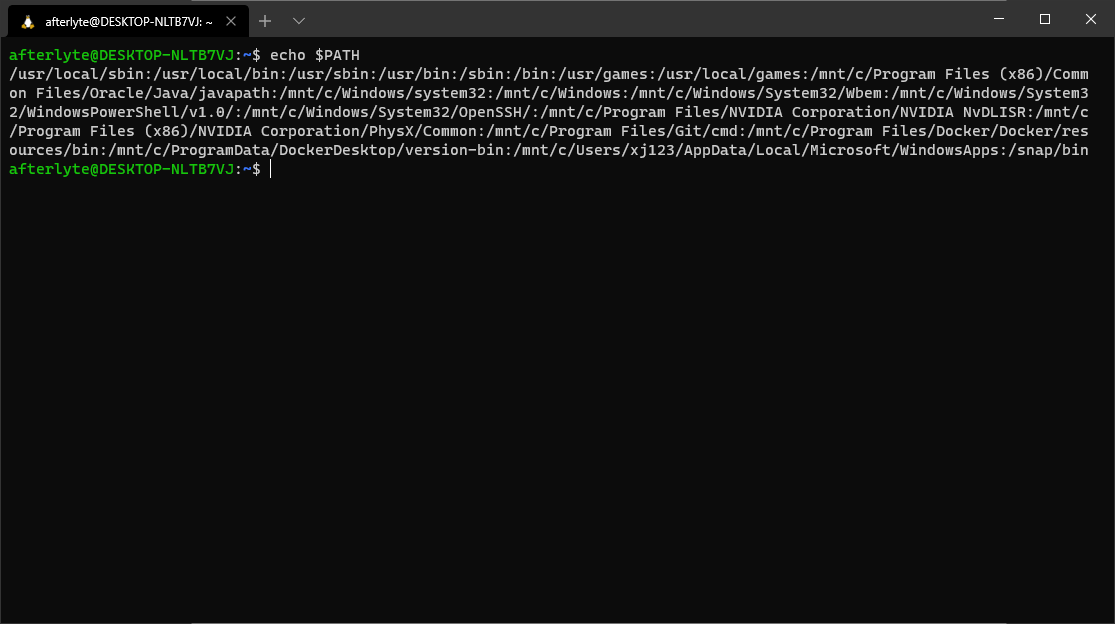
# Environment variables

2. What command will show you all the environment variables? What command will

display the environment variable named **PATH**? Show both with screenshots.

(1 point)

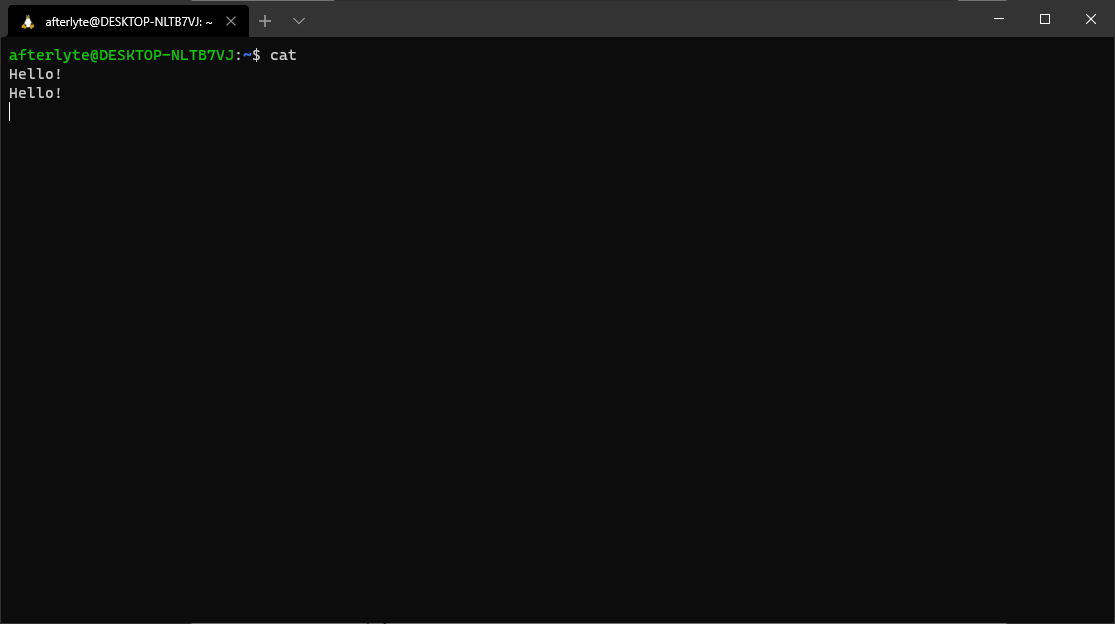




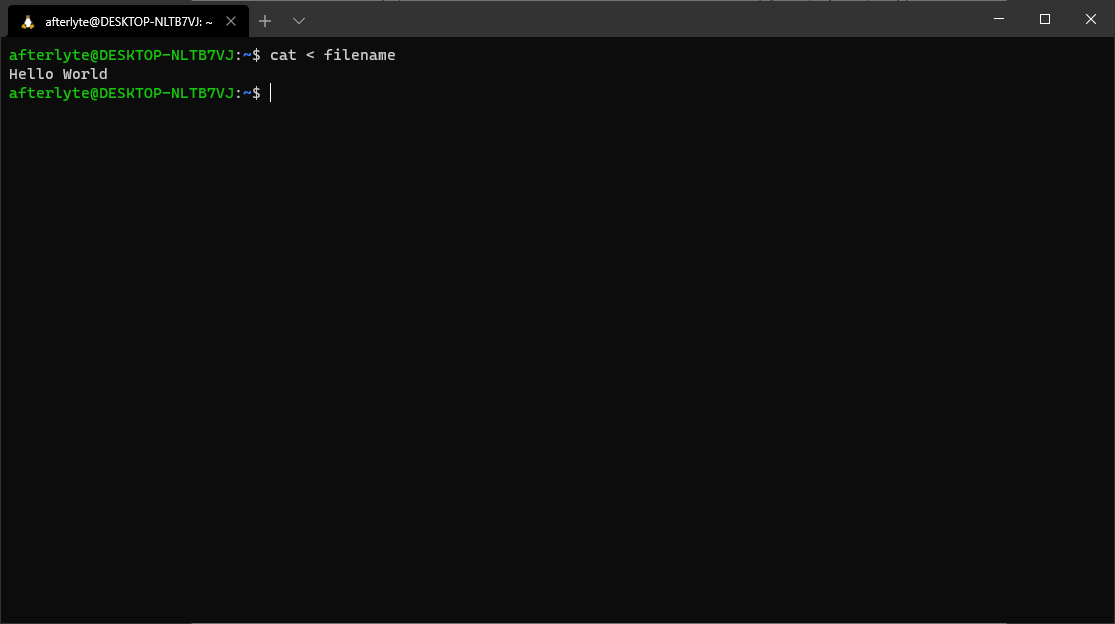
# I/O

1. What are the differences among the following commands? Explain with examples and

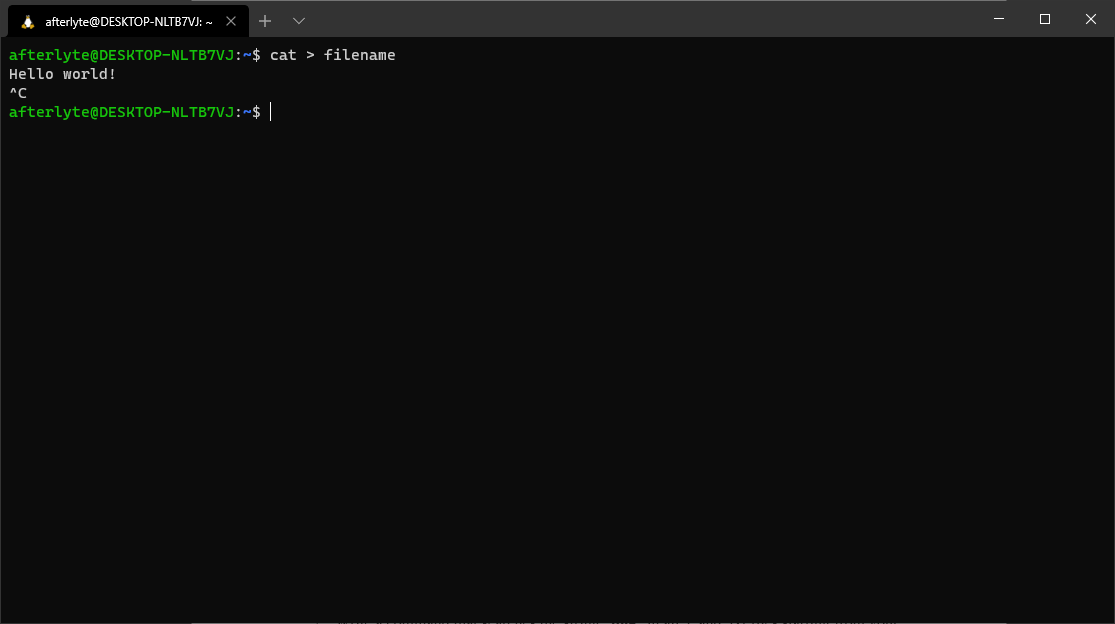
screenshot. ( 4 points, 1 point each) cat – Takes user keyboard input then echoes it back to the terminal



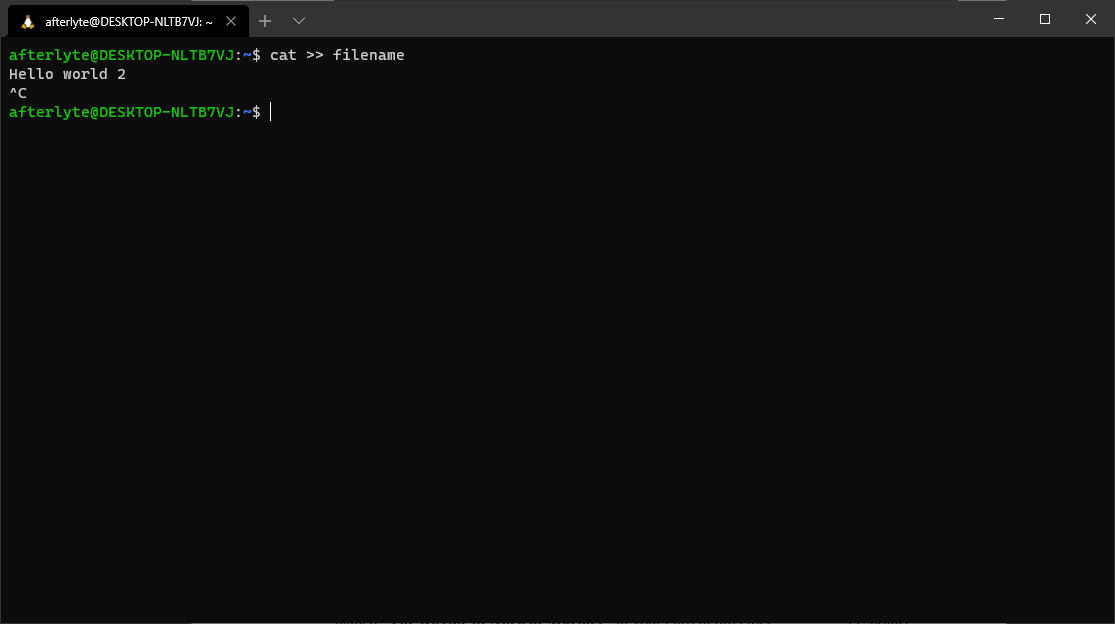
cat < filename – Takes content from filename and prints it to the terminal



cat > filename – Takes user input and replaces previous file content with the input



cat >> filename – same as previous, but appends it to the end instead of replacing





1. Write a command that shows the lines with line numbers the string “**bird**” exists in a file named “**The Rhyme of Ancient Mariner**” in your current directory. (1 point)



1. Write a command that searches the string “**line**” in all .c and .txt files starting from your

current directory and all sub directories. (1 point)



# Metacharacters in Regular Expression

6. What will the following patterns match? Explain. (4 points, 1 point each) a) ^bags$ - this will match explicitly the string “bags”, as it contains a start-of-line and end-of-line character

1. ^…$ - this will match any filename or content in files that has exactly 3 characters (3 single-character matchers with SOL and EOL characters
2. l.g – this will match anything that has an ‘l’ and a ‘g’ at any point with one and only one character in between them
3. ^\. – this will match anything that starts with a literal point (backslash nullifies the typical ‘.’ behavior)

# ‘grep’, ‘find’, ‘pipe’

1. Consider the following file named “**FruitsList.txt**”. Try the following commands and

explain each output with screenshot. ( 6 points, 1 point each)

**FruitsList.txt**

* 1. grep “[A-Z]e” FruitsList.txt

apple

Orange

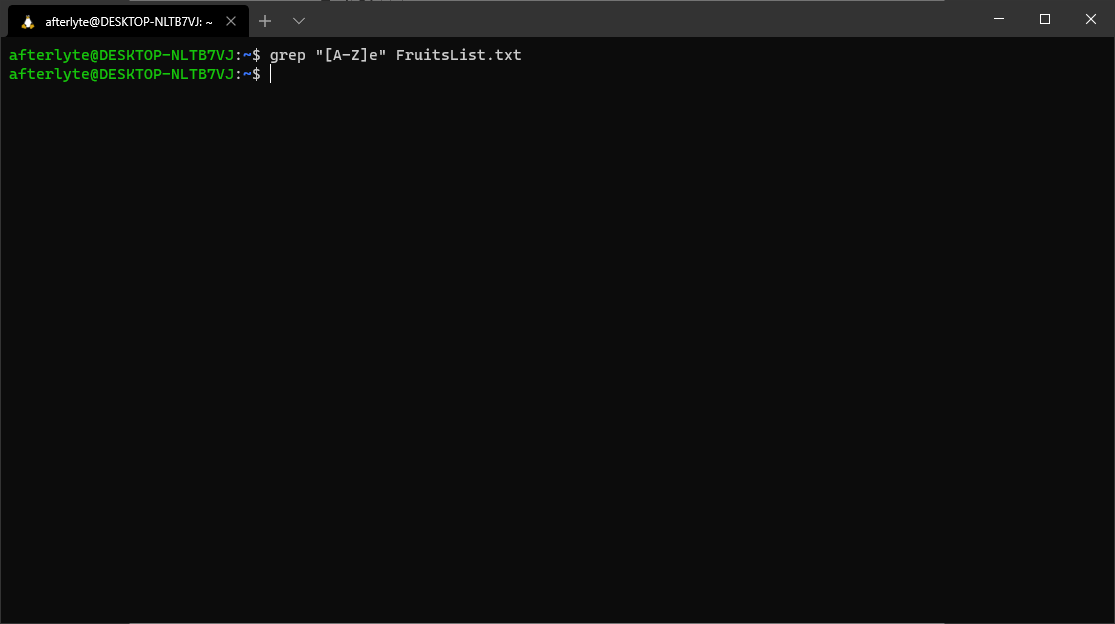
Pineapple

Banana

lemon

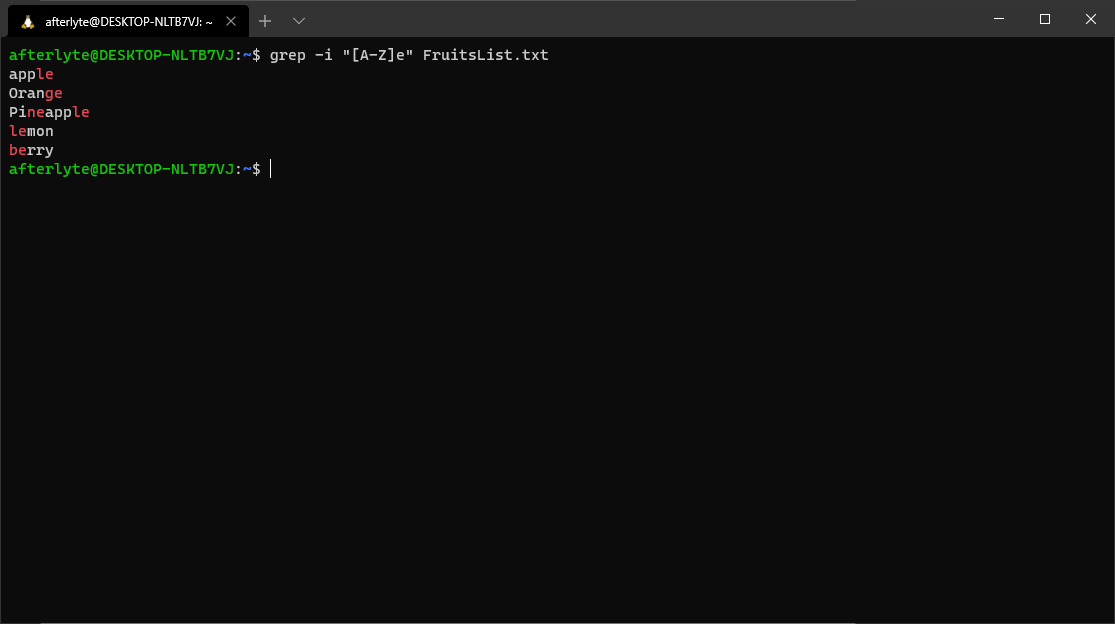
Kiwi

berry



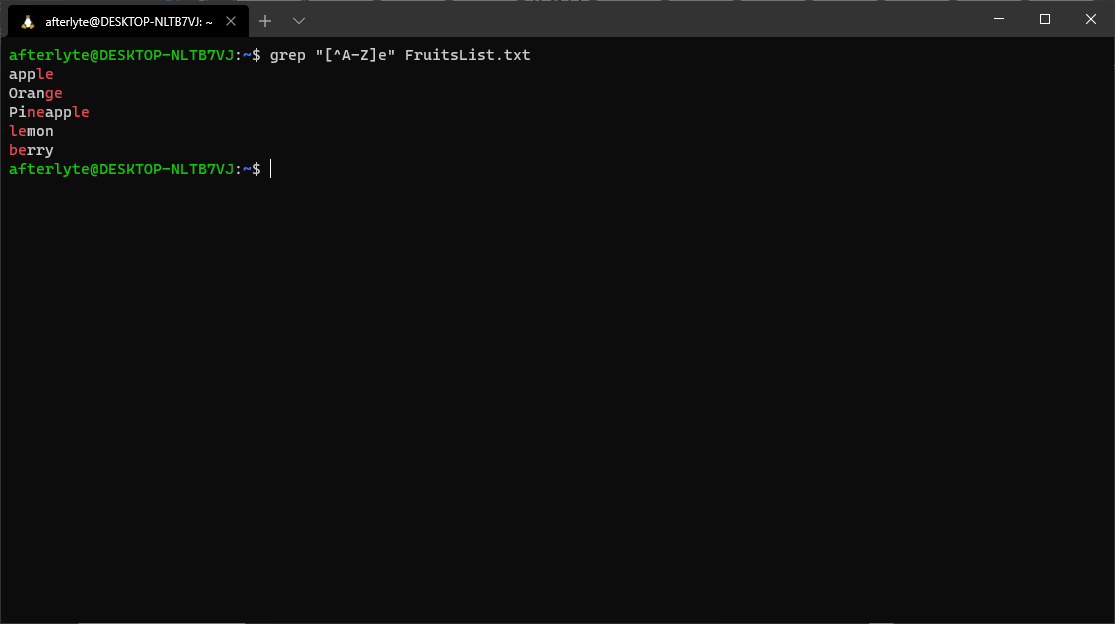
Outputs nothing because there is no instance where a capital letter comes before an ‘e’

* 1. grep -i “[A-Z]e” FruitsList.txt



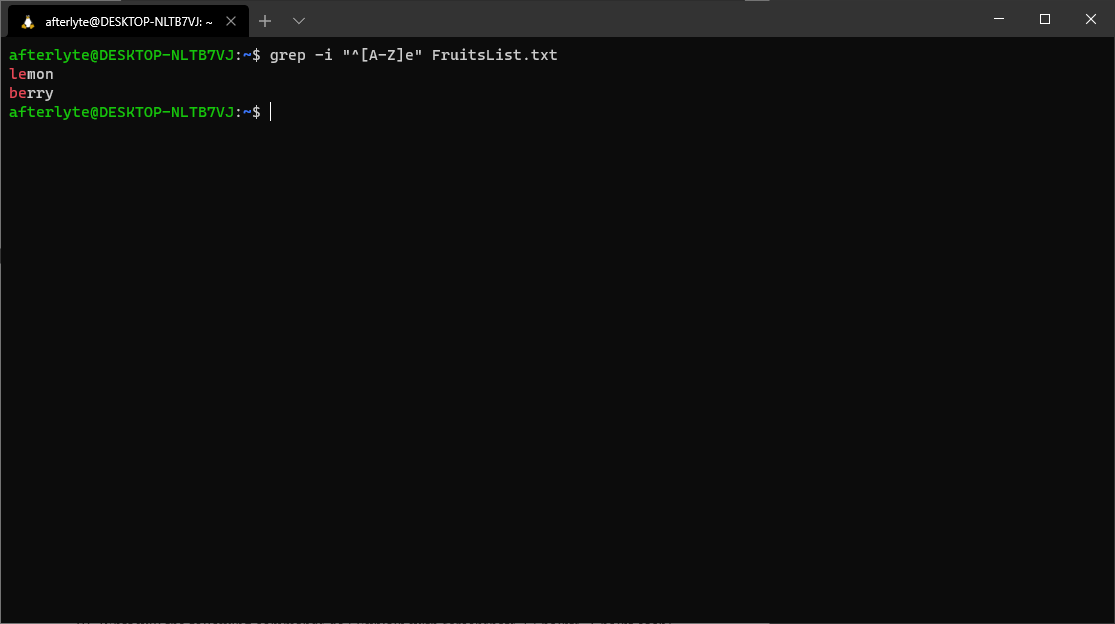
Matched any pattern that had any letter followed by an ‘e’, ignoring case

* 1. grep “[^A-Z]e” FruitsList.txt



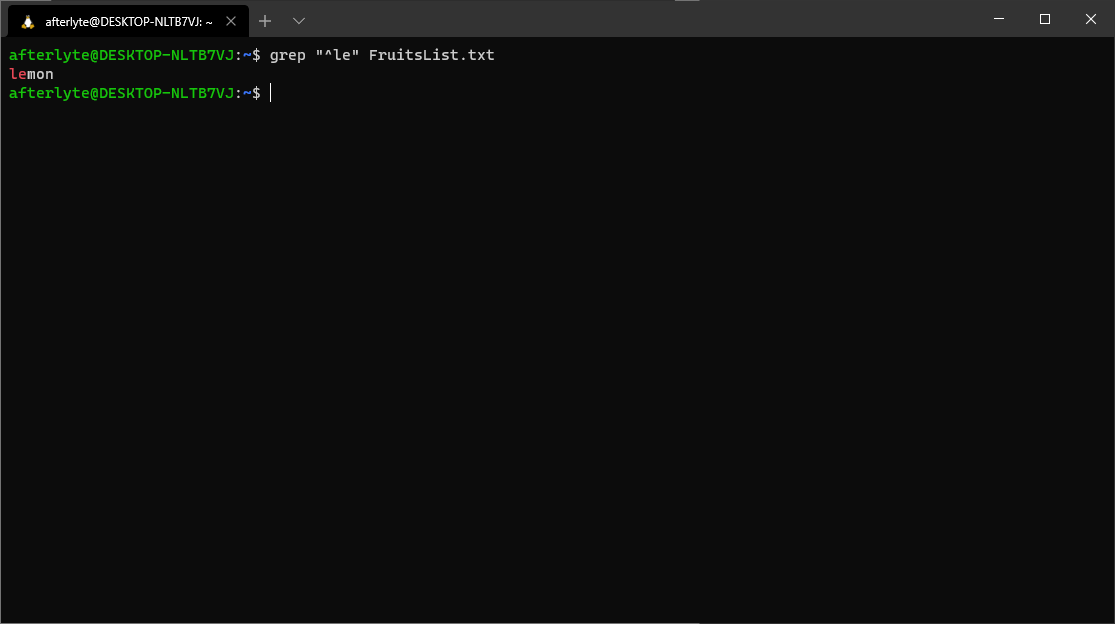
Outputs the same as b), because the carrot symbol causes the search to NOT look for capital letters before an ‘e’, which there are none of

* 1. grep –i “^[A-Z]e” FruitsList.txt



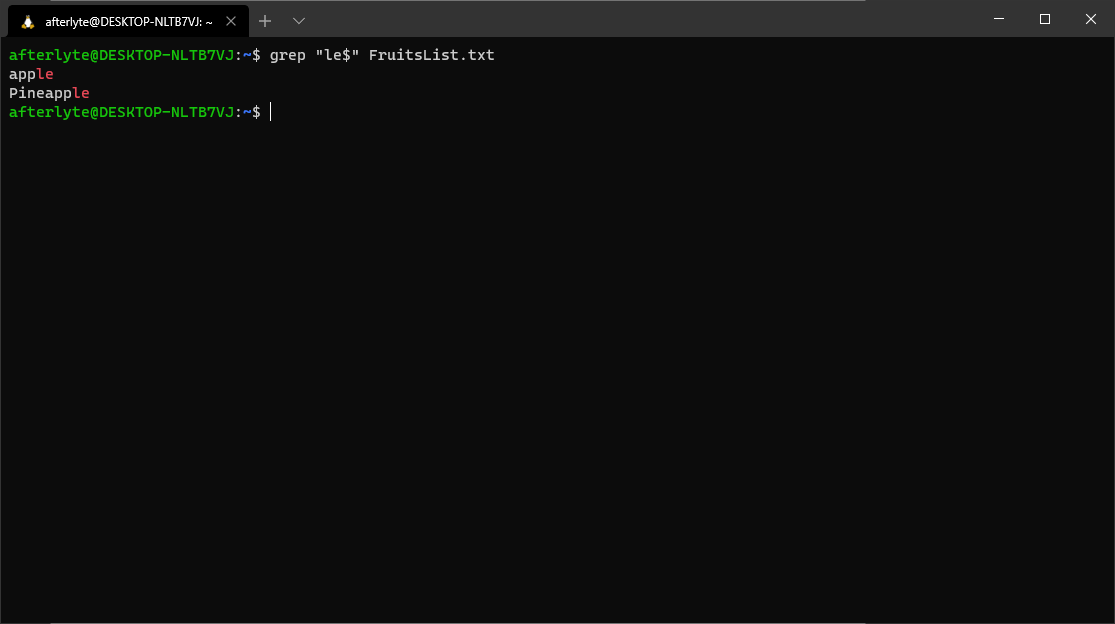
Outputs any instance where the start of a line is any letter (ignore case) followed by an ‘e’

* 1. grep “^le” FruitsList.txt



Outputs any instance where the line starts with ‘le’

* 1. grep “le$” FruitsList.txt



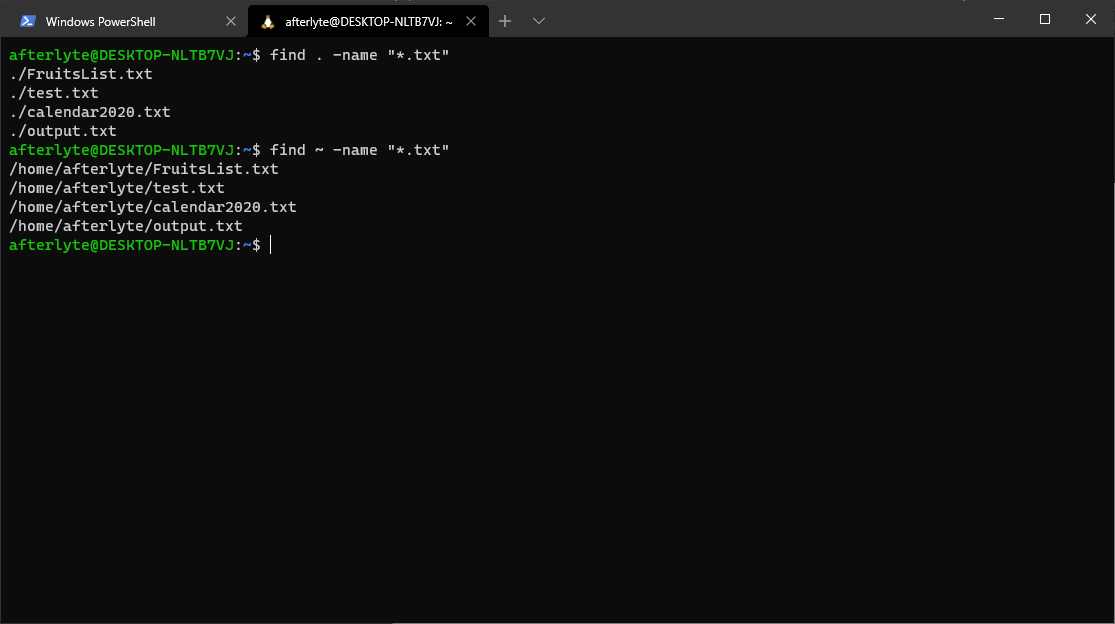
Outputs any instance where the line ends with ‘le’

1. Suppose you are in your home directory. What are the differences between the following

commands? Explain with screenshot. 1 point find . –name "\*.txt"

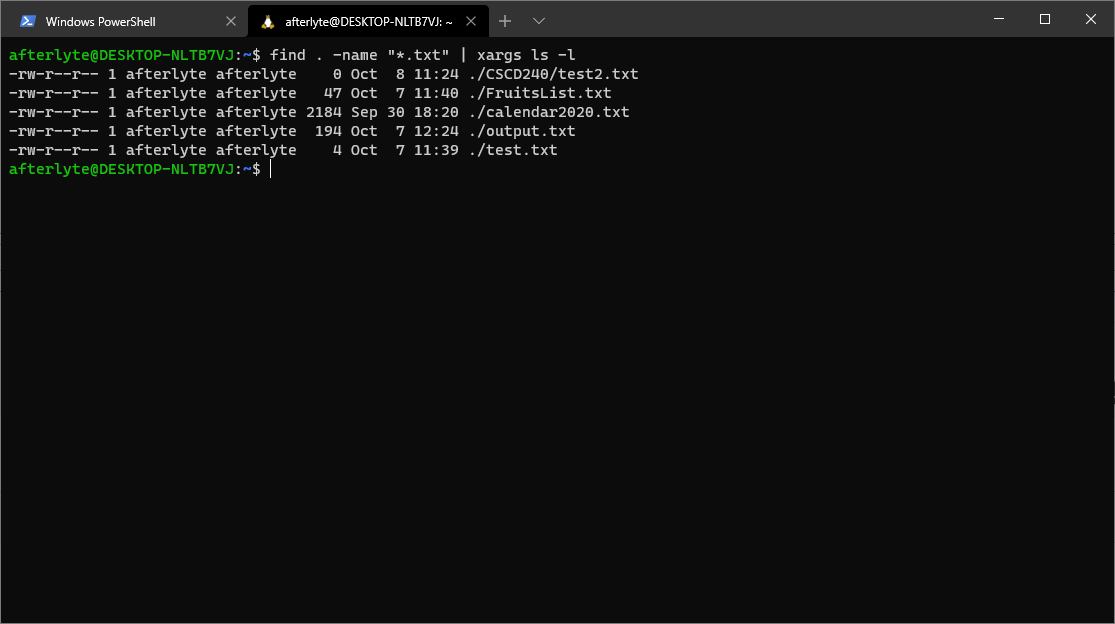
find ~ -name "\*.txt"

The first one shows only the relative path, since that’s what the “.” references, while the second one shows the whole path, since “~” is technically an explicitly defined path



1. Write a command that finds all text files in your home directory and subdirectory and

shows the long listing. 1 point

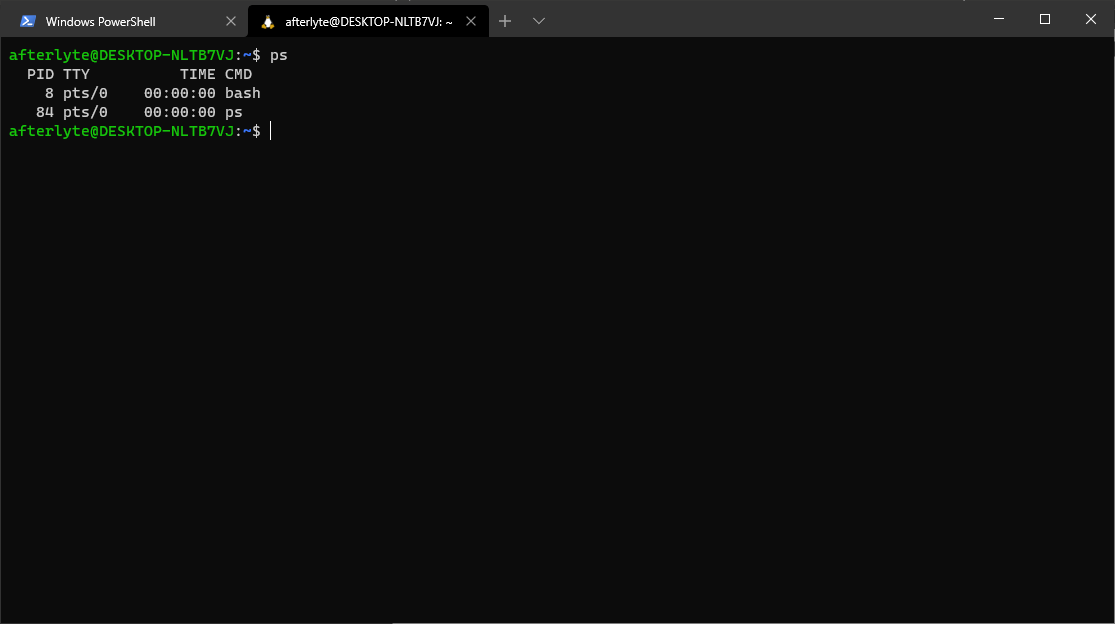


1. What will the following commands do? Explain with screenshots. (2 points, 1 point each)

* 1. ls –l | grep ‘^…….rw’
     1. this will list every file that has at least read and write permissions for other users
  2. grep –n variable \*.[ch]
     1. this will print every line containing the word “variable” in every .c file and .h file, with line numbers

# Processes and Jobs

1. What is process? How will you differentiate processes from jobs? 1 point
   1. Process is a program in execution, and a job is a process that’s attached to a shell session
2. What does **ps** command do? Explain with screenshot.1 point
   1. ps shows all currently running processes



**Submission:**

* A PDF file - Name this file as follows: your last name, first letter of your first name, Lab2.pdf (i.e., YasminSLab2.pdf). This file will contain all your answers. Each question should be copied first and then answered.
* You should turn in through the EWU Canvas system.
* Submission deadline is **Friday, October 9**.
* **No late submission will be accepted**.