**Name:- Sahar Abulikailik Homework(1)**

**Part I: Questions**

1. What is a Graphical User Interface (GUI) and how does that differ from the Command Line Interface (CLI)?

A Graphical User Interface (GUI) is a type of user interface that allows users to interact with the computer through graphical elements such as icons, buttons, and windows. A GUI presents information visually and allows users to manipulate data using a pointing device such as a mouse. But it doesn’t explore all the option user can have to interact with the computer.

A Command Line Interface (CLI) is a type of user interface that allows users to interact with electronic devices through text commands. CLI is typically used in text-based interface and allows users to type commands at a command prompt. Unlike GUI, CLI does not use visual elements and users need to memorize commands and their syntax to use the system but still is more powerful than the GUI.

In short, GUI is more user-friendly and easy to use while CLI is more powerful and efficient but requires more skill and knowledge to use effectively.

1. What does the shell do?

The shell is a program that lets you interact with an operating system using commands.

1. What is the command used for listing things in a directory?

ls

1. What is the command used for changing directories?

cd

1. What command would you use to get your current working directory?

PWD

1. How do you get the manual for these commands?

Man

1. What does the shell prompt look like?

Dollar sign

1. How would you list things in a directory in chronological order?

Ls -t

1. Name two ways to get to the ‘home’ or ‘root’ directory.

~

Cd

Cd /home/

1. What is the difference between an absolute path and a relative path?

An absolute path is a complete path to a file or directory starting from the root directory.

A relative path is a path to a file or directory relative to the current working directory.

1. What are the two relative path directories we talked about and what do they mean?

./dir : this is the directory we are currently in (current directory)

../dir: this is a directory that is level above the current working directory.

1. Why shouldn’t you put spaces in filenames?

Because the first name will be interpreted as an argument and the second part will be the incorrect file name.

1. Name a way to have multiple readable words in a filename without spaces?
2. CamelCase, where the first letter of each word is capitalized.
3. Underscore (**\_**)
4. What is the program nano, what does it do? Do you use nano? If not what do you use?

Is a text editor. It provides a simple interface for editing text files and has basic features such as search and replace, cut, copy and paste, etc.

I never used it before this class because am new to linux.

1. \* is a form of a wildcard. What does it mean?

ALL the files that have the same condition

1. The following code would match what? ls \*.txt

All the files that have the extension .txt

1. Why do we need to be careful with the mv command?

Because it can overwrite an existing file or directory if the destination already exists and you are moving or renaming a file or directory to that location.

1. Why do we need to be careful with the rm command?

Because it deletes files and directories permanently, and they cannot be recovered once they have been deleted.

1. What is the difference between > and >>?

**>** it redirects output to a file, and it overwrites the contents of the file if it already exists.

**>>** it is used to append output to a file, adding the new output to the end of the file if it already exists

1. What does head do?

It displays the first specified number of lines from a file

1. What is the purpose of | (pipe)?

It is used to redirect the output of one command as the input to the next command after the pipe.

1. Interpret the following command:

cat huge\_file.fasta | uniq | head -n 5 >proteins.fasta

cat huge\_file.fasta reads the contents of huge\_file.fasta and then feeds that as input to next command where uniq filters out duplicate lines, then head -n 5 selects the first 5 lines and finally, >proteins.fasta writes the result to a file named proteins.fasta

1. What is a loop and when would you use it?

A loop is a control structure in programming that allows repeating a block of code multiple times. It is used to automate repetitive tasks and make the code more efficient.

For thing in things

Do

Something

Done

1. In a directory with the following files. What would the following loop do?

$ for filename in \*.txt; do

> cat ${filename} >>fruits.txt

> done

This loop is written in shell script, and it will concatenate the contents of all files with the **.txt** extension in the current directory into a single file named **fruits.txt**.

1. What is a shell script?

A shell script is a program written in a shell programming language that can be executed in a Unix-like operating system to automate repetitive tasks and perform complex operations. It is a series of commands saved in a file that can be executed automatically.

1. What are the benefits of writing and running a script over typing the code in?

To automate repetitive tasks and being able to handle errors at the same time, and also it can be reused and improved.

1. When looking at a script what does # mean? Why would you use one?

It means that the text after this sign is not executable but used for guidance and comments.

1. What does grep stand for and what does it do?

**grep** is "global regular expression print". It is used to search for a specified pattern in one or more files and return the lines that match the pattern.

1. What does find do?

**find** is used to search for files and directories in a specified location based on various criteria, such as name, size, type, modification time, and others.

## Part II - Writing Code

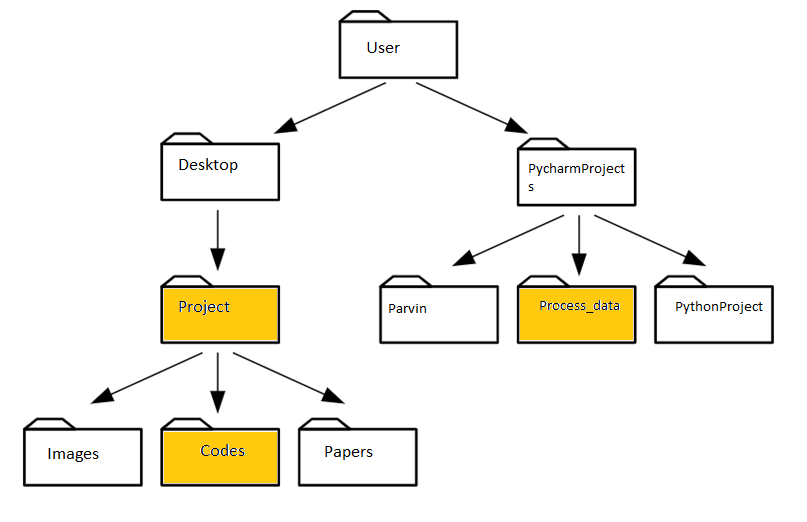
As part of this exercise please open the shell and practice moving around into different files. Best practice is to try a little bit every day. Try to challenge yourself by not using the mouse. For each of the questions below type your code and the result from the shell prompt.

1. Open the shell and change directories to the desktop and list the contents, paste your code here.

cd ~/Desktop

ls

1. Draw the file structure from the root directory into one of the folders on the desktop. You can submit a pdf with a drawing. Select three locations and type the commands for changing directories to those locations use a combination of absolute and relative paths.



cd /Users/User/PycharmProjects/process\_data

cd ../../Desktop/Project

cd ./Project/Codes

1. On your computer create a directory on your desktop. Create two files without opening them that have file extensions .txt. Then use wildcards to list the files in that directory. Paste you code here.

# create a directory on the desktop

mkdir ~/Desktop/new\_dir

# create two files in the directory

touch ~/Desktop/new\_dir/file1.txt ~/Desktop/new\_dir/file2.txt

# use wildcards to list the .txt files in the directory

ls ~/Desktop/new\_dir/\*.txt

1. Download the file [Hutia\_DNA.fasta](https://www.dropbox.com/s/07y610c8zt63nt5/Hutia_DNA.fasta?dl=0) from the Github repo. Make a new directory on the desktop. Move the file into that directory. List the contents of the directory. How many lines are in this file? Fasta files are a type of data file that holds DNA sequences. They are all formatted the same. The first line starts with a > and a name followed by a line break, and then DNA sequence followed by a line break. The next line starts with an > and another name followed by a line break and the DNA sequence. With that in mind, how could you use linux commands to determine how many sequences are in this file? How many sequences are in this file?

# create a new directory on the desktop

mkdir ~/Desktop/Hutia\_DNA

# move the file from the Downloads folder into the new directory

# worked well on powershell

mv ~/Downloads/Hutia\_DNA.fasta ~/Desktop/Hutia\_DNA

# list the contents of the directory

ls ~/Desktop/Hutia\_DNA

# To determine the number of lines in the file

wc -l ~/Desktop/Hutia\_DNA/Hutia\_DNA.fasta

Answer: 2382004

#To determine the number of sequences in the file

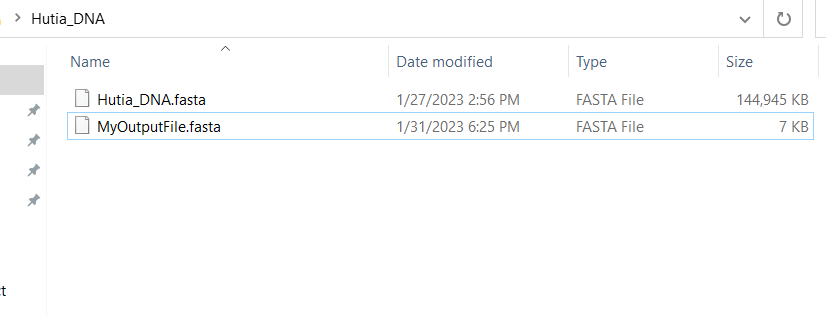
grep -c ">" Hutia\_DNA.fasta

1. Build a single line of code that would take the first 100 sequences of this file and put them in a new file called ‘MyOutputFile.fasta’. Use a combination of wc, sort and uniq in a single line of code to tell me how many unique lines are in this file. Paste your commands and the answer here.

head -100 Hutia\_DNA.fasta > MyOutputFile.fasta; wc -l MyOutputFile.fasta | awk '{print $1}'

Answer

100



1. Write code that would create a loop to copy all the files in one directory ending in .txt to another directory. Paste the code here.

#!/bin/bash

# Loop through all the files in the source directory

for file in $~/Desktop/\*.txt

do

cp $file $~/Desktop/Hutia\_DNA

done

1. Write a bash script with in-line documentation (hint #) to show how to run a made up python program (script) called 'find\_taxa.py' on set of files ending in ‘.fasta’. Show in the script how to run the program on each of those files. What is the name of your script? Type the script here.

#!/bin/bash

# This script runs the 'find\_taxa.py' program on all files ending in '.fasta' in the current directory.

# Loop through all the files in the current directory (Hutia\_DNA)

for file in \*.fasta; do

# Run the 'find\_taxa.py' program on each '.fasta' file

python find\_taxa.py "$file"

done

chmod +x run\_find\_taxa.sh

1. From the Hutia\_DNA.fasta file tell me how on how many lines do we find the pattern ‘GAGA’. What was the code used to find this?

grep -c 'GAGA' Hutia\_DNA.fasta