1. What is a Graphical User Interface (GUI) and how does that differ from the Command Line Interface (CLI)?
   1. GUI is a form of communication by using a mouse and menu technique that differs from CLI because it is a scripting language and automatically performing repetitive tasks.
2. What does the shell do?
   1. Shell allows us to write scripts by using commands to create programs
3. What is the command used for listing things in a directory?
   1. ls
4. What is the command used for changing directories?
   1. cd [type directory]
   2. cd ..
5. What command would you use to get your current working directory?
   1. pwd
6. How do you get the manual for these commands?
   1. man
7. What does the shell prompt look like?
   1. Text

      Description automatically generated
8. How would you list things in a directory in chronological order?
   1. ls -F
   2. sort
9. Name two ways to get to the ‘home’ or ‘root’ directory.
   1. cd
   2. cd ~
10. What is the difference between an absolute path and a relative path?
    1. Absolute paths have the root element and complete sections/directory list to the location or file
    2. Relative is where I am (directory, file, etc.)
11. What are the two relative path directories we talked about and what do they mean?
    1. “..” referes to the parent directory and “.” Is the current directory you are in
12. Why shouldn’t you put spaces in filenames?
    1. It signifies the end of the file
13. Name a way to have multiple readable words in a filename without spaces?
    1. Underscores, numbers, letters, upper case letter
14. What is the program nano, what does it do? Do you use nano? If not what do you use?
    1. Nano is a text editor, I prefer BBEdit.
15. \* is a form of a wildcard. What does it mean?
    1. It matches characters based off a pattern or similar wording or text files.
16. The following code would match what? ls \*.txt
    1. Anything with .txt
17. Why do we need to be careful with the mv command?
    1. We are changing the file name
18. Why do we need to be careful with the rm command?
    1. Deleting is forever
19. What is the difference between > and >>?
    1. > redirects the commands output to a file
    2. >> if the file does not exist, it will create the file
20. What does head do?
    1. Displays the first lines of a file
21. What is the purpose of | (pipe)?
    1. It’s to use the output of the first command to be implicated into the input of the second
22. Interpret the following command:
    1. Redirects and creates the 5 first lines and creating a new file called proteins.fasta

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

1. What is a loop and when would you use it?
   1. A loop is a construct that slows you to repeat a command for an item in a list.
2. In a directory with the following files. What would the following loop do?
   1. It will add fruits.txt file into another folder
   2. apple.txt, banana.txt, orange.txt $ for filename in \*.txt; do > cat ${filename} >>fruits.txt > done
3. What is a shell script?
   1. A program! Small or big
4. What are the benefits of writing and running a script over typing the code in?
   1. Saves time
5. When looking at a script what does # mean? Why would you use one?
   1. You put # so that you can keep tract of each section of the shell script by writing notes and the program will not consider it while running
6. What does grep stand for and what does it do?
   1. Grep searches any input files that have a matching of one or more patterns
7. What does find do?
   1. Descends the directory tree for each path listed and evaluates the expression in terms of each file in the tree

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.
   1. mv data-shell/ /Users/alexciton/Desktop
2. 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.
3. 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.
   1. ls \*.txt
4. Download the file Hutia\_DNA.fasta 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?**
   1. 2382004 2382004 148423288 Hutia\_DNA.fasta
   2. We can use a wildcard to find out how many sequences there are in this fasta file or use grep -c
   3. 1191002 sequences
5. 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.
   1. Cat Hutia\_DNA.fasta | wc -l | sort | uniq -c | head -n 100 1> MyOutputFile.fasta
6. 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.
   1. For datafile in \*.txt

do

cat $datafile >> newdirectory/

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.
   1. for file in “$@” #”$@” takes into account any file and infinite amount of files, in this case it is a fasta file

do

echo “find taxa in $file:”

done

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?
   1. grep “GAGA” Hutia\_DNA.fast
   2. although, every time I would attempt this, it would not work. Also, when I just use grep -c, it is not working anymore. I must of done something to the fasta file maybe?