**Assignment 2**

**Aim:** The purpose of this assignment is to practice terminal commands and navigation

**2.1 Introduction**

The purpose of this assignment is to familiarize you with using the terminal.  
Your terminal session will be logged, so refrain from typing sensitive information. Stay on task and use the session for its intended purpose.

For any questions, answer directly in the terminal by "echoing" your response as a string:

echo "This is my response"

**2.2 Instructions**

**Part 1: Log in to the SSH Session**

1. Open the terminal using Ctrl + T.
2. Log in to the following SSH session:
   * Host: en4xxxxxxl.tail8bc7c.ts.net
   * Port: 22xx
   * Username: Your <> ID.
   * Private Key: Use the key created in **Assignment 1** as your identity file.

If the key is missing, contact Dr. Aukes for a temporary password.

1. Start saving your session using the following command:

script -q -f -T log1.time log1.session

**Part 2: Filesystem Navigation**

1. Navigate to the filesystem’s root directory: cd /
2. List all files in the root directory: ls
3. Navigate to your home directory using the special character.
4. List all files in the root directory *from* your home directory.
5. List only the files in your directory that start with "ap".
6. Answer: What does cd do without arguments? Use the following command to answer:

echo "This is my response"

**Part 3: Finding and Grep Commands**

1. Find all files in your user folder that end with ".txt" (case-insensitive).
2. Find all files in your user folder that start with "b" (case-insensitive).
3. Use grep to list all files in your user folder containing the word "broccoli" (case-insensitive).
4. Use grep to list all files containing "broccoli" with associated lines and line numbers.

**Part 4: Directory and File Operations**

1. Create the following nested directory structure using mkdir and touch:

temp/

├── folder-a

│ ├── file-a.txt

│ ├── file-b.txt

├── folder-b

│ ├── file-c.txt

│ └── file-d.txt

* Question: Which mkdir flag reduces the number of commands required to make a nested structure?

1. Navigate to /temp/folder-b/ and complete the following tasks:
   * Write the string **"The quick brown fox jumped over the lazy dog"** to a file (file-e.txt) using >.
   * Display the contents of file-e.txt.
   * Append a second string to the file and re-display the contents.
   * Use another command to append (not replace) the contents and re-display them.
   * Copy file-e.txt to file-f.txt and display its contents.
   * Write a string to file-g.md using > and display its contents.
   * Question: What happened to the original contents of file-f.txt? Use the echo command to explain.

**Part 5: Advanced Functions**

1. Show the total disk usage of the temp/ directory in human-readable format.
2. Replace all instances of the word "stew" with "pot" in files within the user folder.

**Part 6: Variables and Environment Variables**

1. Assign a string to a variable using the equal sign:

my\_variable="This is my string"

1. Print the value of the variable.
2. Embed the variable in another string and echo it.
3. Export the variable as an environment variable.
4. Identify all environment variables using a Bash function.

**Part 7: Processes**

1. Launch a process:

sleep infinity &

1. Display all running processes (not limited to the current user or terminal session).
2. Find the sleep process ID (PID) using pipes and grep.
3. Kill the sleep process using its PID.

**Part 8: Submission**

1. Copy and paste the entire Bash session (including errors) to a new text editor.