ICS 2305 SYSTEMS PROGRAMMING

TAKE-AWAY CAT

Name: AKECH DAU ATEM

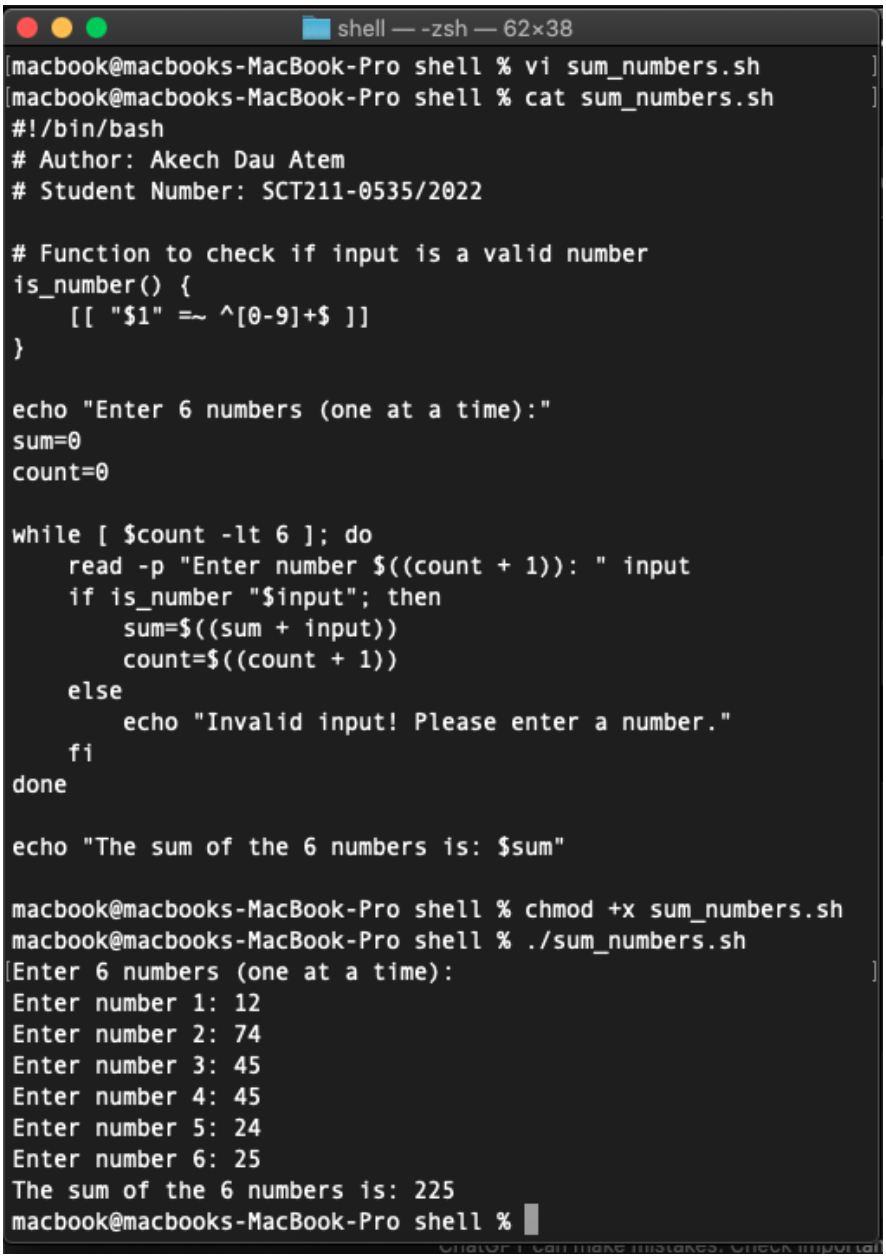
RegNo: SCT211-0535/2022

Please note that as I am using macOS, some of the code may be Unix-dependent and there could be compatibility issues when running the .sh files on other systems. I have included the outputs to ensure clarity.

Thank you for your understanding.

Question 1

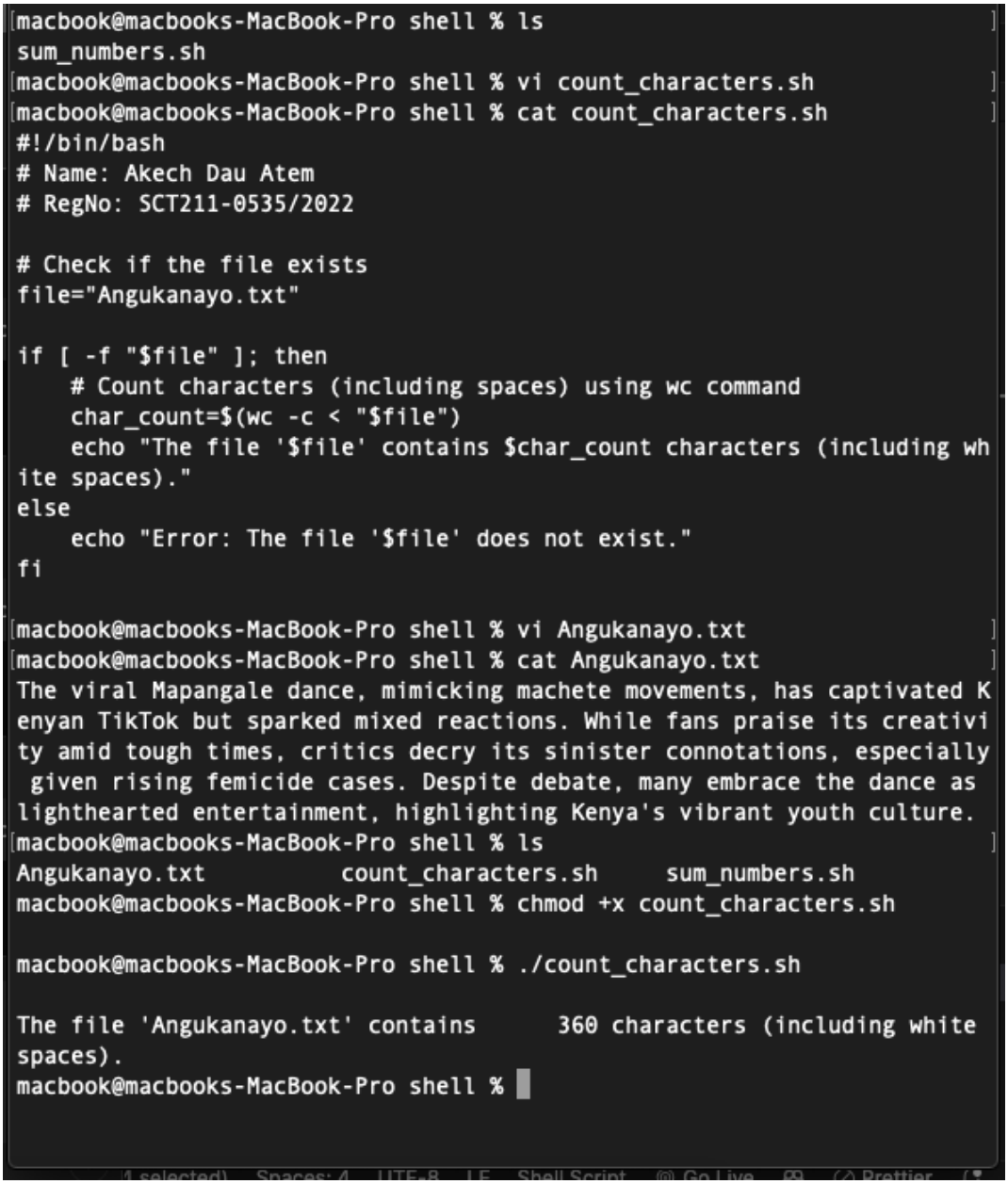
Write the shell script that reads any 6 numbers from the keyboard and prints their sum. If non-numbers are entered, it throws an alert to key in numbers only.



Question 2

Part (a): Counting Characters in a Text File

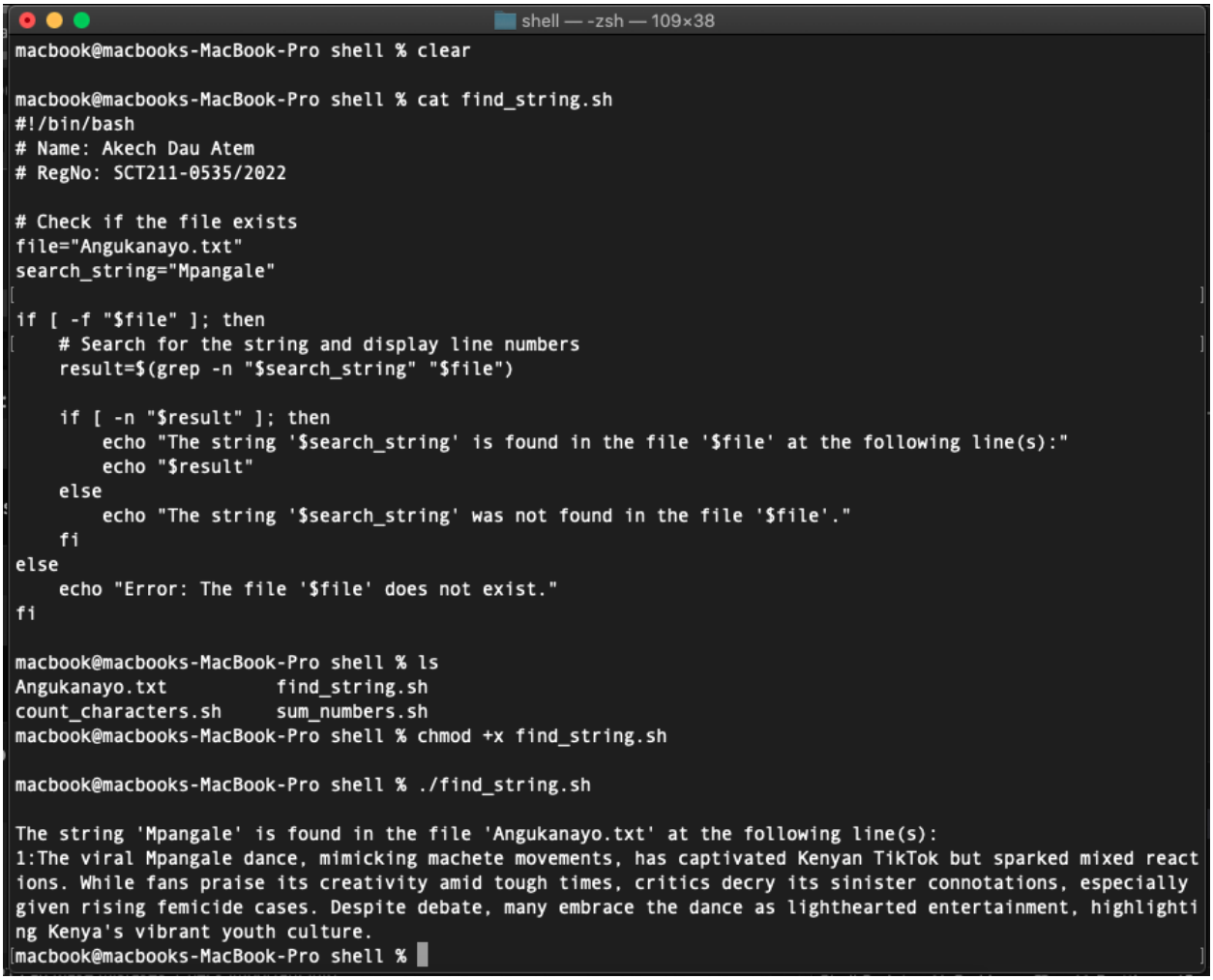
Write a shell script that reads the text in a file called Angukanayo.txt and outputs the number of characters (inclusive of white spaces) in the file.



```
macbook@macbooks-MacBook-Pro shell % ls
sum_numbers.sh
macbook@macbooks-MacBook-Pro shell % vi count_characters.sh
macbook@macbooks-MacBook-Pro shell % cat count_characters.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Check if the file exists
file="Angukanayo.txt"

if [ -f "$file" ]; then
    # Count characters (including spaces) using wc command
    char_count=$(wc -c < "$file")
    echo "The file '$file' contains $char_count characters (including wh
ite spaces)."
else
    echo "Error: The file '$file' does not exist."
fi

macbook@macbooks-MacBook-Pro shell % vi Angukanayo.txt
macbook@macbooks-MacBook-Pro shell % cat Angukanayo.txt
The viral Mapangale dance, mimicking machete movements, has captivated K
enyan TikTok but sparked mixed reactions. While fans praise its creativi
ty amid tough times, critics decry its sinister connotations, especially
 given rising femicide cases. Despite debate, many embrace the dance as
lighthearted entertainment, highlighting Kenya's vibrant youth culture.
macbook@macbooks-MacBook-Pro shell % ls
Angukanayo.txt          count_characters.sh     sum_numbers.sh
macbook@macbooks-MacBook-Pro shell % chmod +x count_characters.sh

macbook@macbooks-MacBook-Pro shell % ./count_characters.sh

The file 'Angukanayo.txt' contains      360 characters (including white
spaces).
macbook@macbooks-MacBook-Pro shell %
```

Part (b): Searching for a String in the Text File

Write a script to report if the file Angukanayo.txt contains the string Mpangale and the line number(s) on which it occurs



Question 3

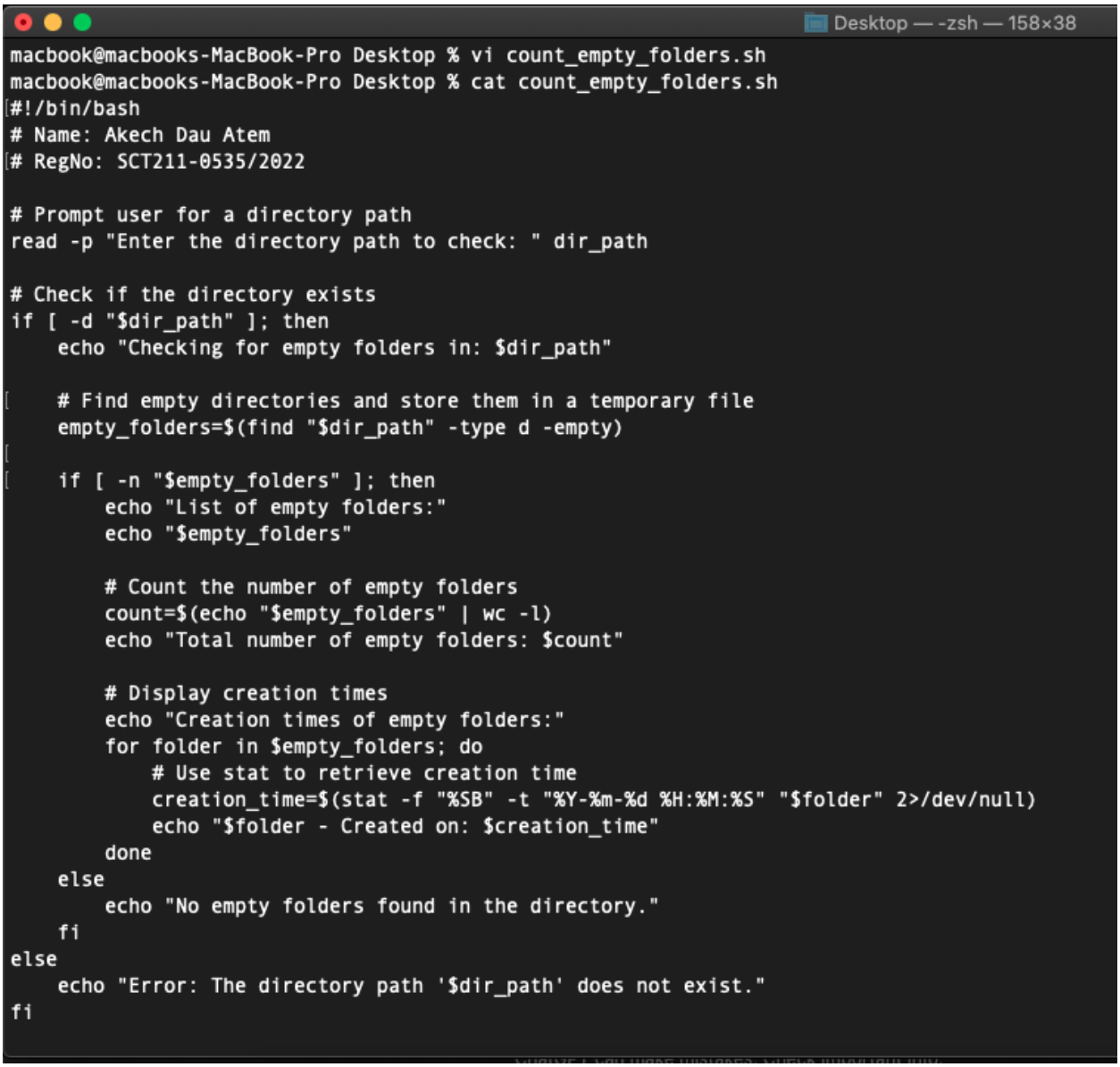
Task

Write a shell script that, given a person's UID, tells how many times that person is logged on to the system and the time(s) of their login.

Question 4

Task

Write a shell script that counts the number of empty folders in a directory (size = 0) and outputs the list of those folders along with their creation time.



1. **Directory Check**:

The script checks if the provided directory exists using -d.

2. **Find Empty Folders**:

   The find command searches for directories (-type d) that are empty (-empty).

3. **Count and List Empty Folders**:

   If empty folders are found, they are listed, and their total count is displayed using wc -l.

4. **Retrieve Creation Times**:

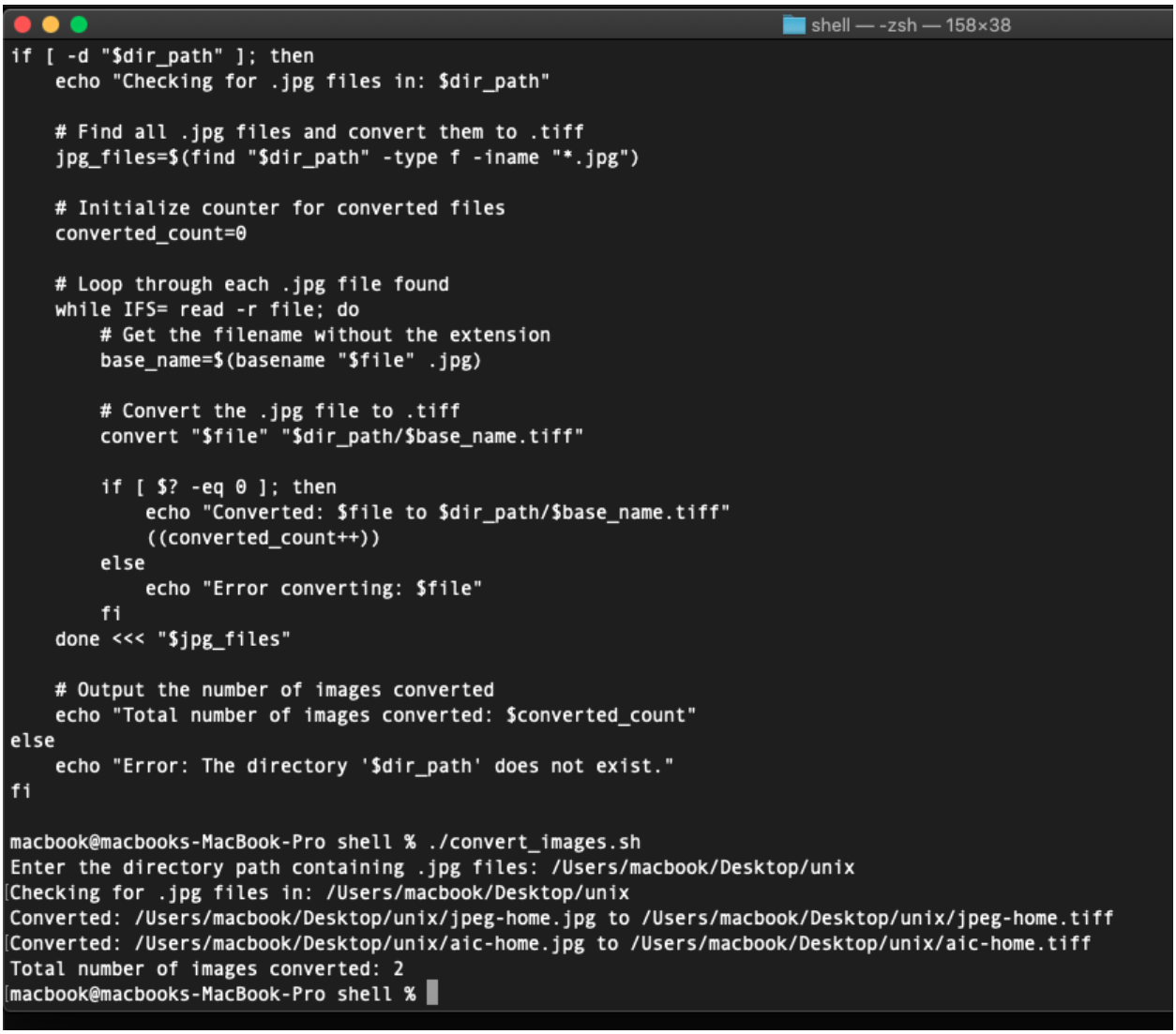
   The stat command is used to retrieve the creation time of each empty folder (macOS-specific).

```
macbook@macbooks-MacBook-Pro Desktop % chmod +x count_empty_folders.sh

macbook@macbooks-MacBook-Pro Desktop % ./count_empty_folders.sh

Enter the directory path to check: /Users/macbook/Desktop
Checking for empty folders in: /Users/macbook/Desktop
List of empty folders:
/Users/macbook/Desktop/pop
/Users/macbook/Desktop/SWMS/env/include/python3.12
/Users/macbook/Desktop/processes/.dist
/Users/macbook/Desktop/file_management/.dist
/Users/macbook/Desktop/env/include/python3.12
/Users/macbook/Desktop/New Folder With Items/portfolio/.git/objects/info
/Users/macbook/Desktop/New Folder With Items/portfolio/.git/refs/tags
/Users/macbook/Desktop/unix
/Users/macbook/Desktop/.venv/include/python3.12
/Users/macbook/Desktop/stack/.dist
/Users/macbook/Desktop/c/.dist
Total number of empty folders:       11
Creation times of empty folders:
/Users/macbook/Desktop/pop - Created on: 2024-11-26 15:22:40
/Users/macbook/Desktop/SWMS/env/include/python3.12 - Created on: 2024-11-18 10:01:59
/Users/macbook/Desktop/processes/.dist - Created on: 2024-11-21 12:54:38
/Users/macbook/Desktop/file_management/.dist - Created on: 2024-10-28 09:08:46
/Users/macbook/Desktop/env/include/python3.12 - Created on: 2024-11-18 09:59:27
/Users/macbook/Desktop/New - Created on:
Folder - Created on:
With - Created on:
Items/portfolio/.git/objects/info - Created on:
/Users/macbook/Desktop/New - Created on:
Folder - Created on:
With - Created on:
Items/portfolio/.git/refs/tags - Created on:
/Users/macbook/Desktop/unix - Created on: 2024-11-26 05:21:57
/Users/macbook/Desktop/.venv/include/python3.12 - Created on: 2024-06-19 14:37:31
/Users/macbook/Desktop/stack/.dist - Created on: 2024-10-15 06:41:16
/Users/macbook/Desktop/c/.dist - Created on: 2024-10-11 18:15:58
macbook@macbooks-MacBook-Pro Desktop %
```

Question 5

Task

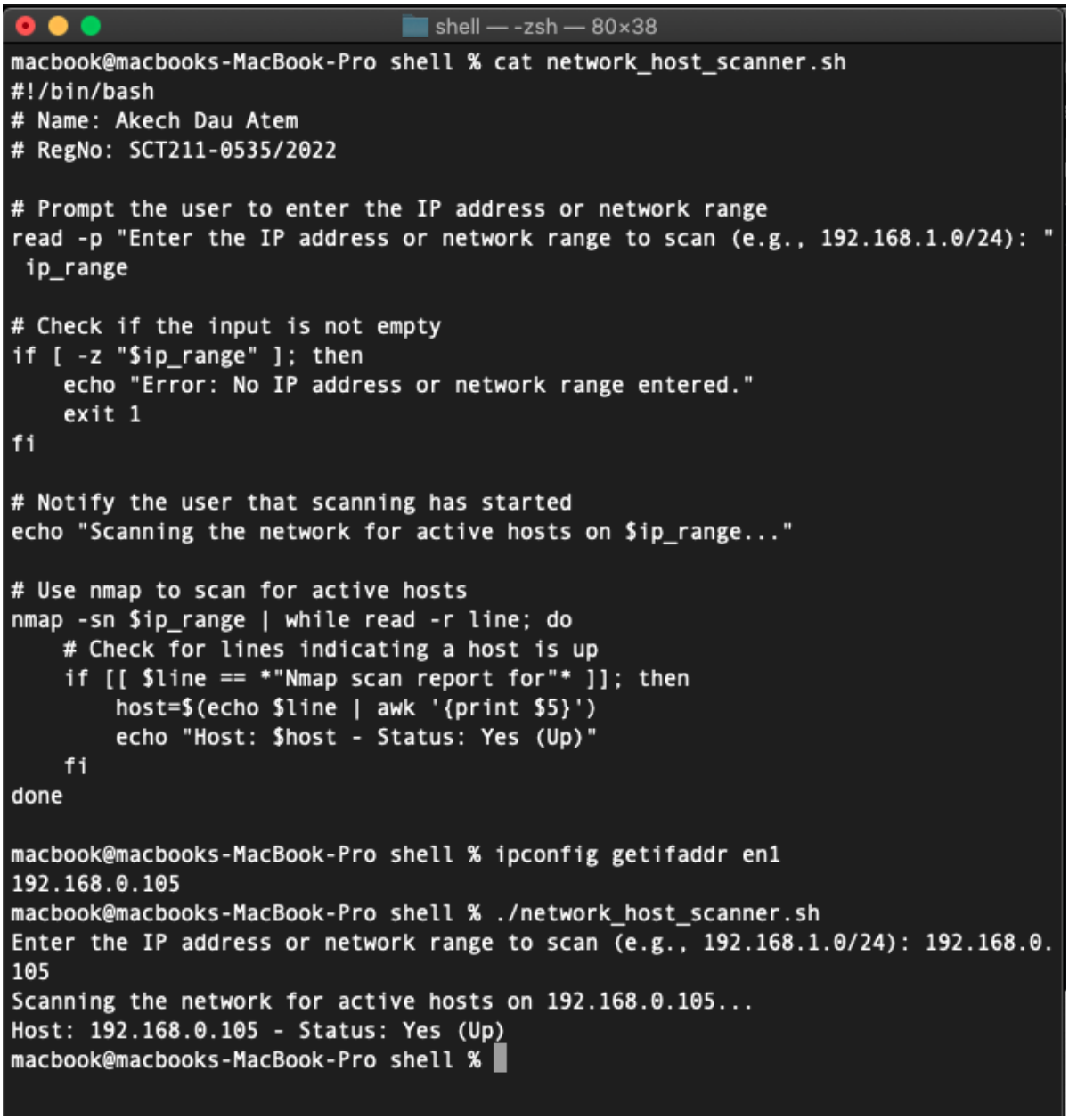
Write a shell script that will convert all .jpg files in a given directory into .tiff files. The user should be prompted to enter the file path. The output after conversion should give the number of images converted.



Question 6

Task:

Write a bash script that scans the network for hosts attached to an IP address. The script should show if the host is up, indicating "Yes" if up.



Question 7

Task:

The Department Chairman of Computing has requested a mail merge service to invite students to a webinar. Write a shell script that reads a text file with names and emails, then sends personalized invitations.

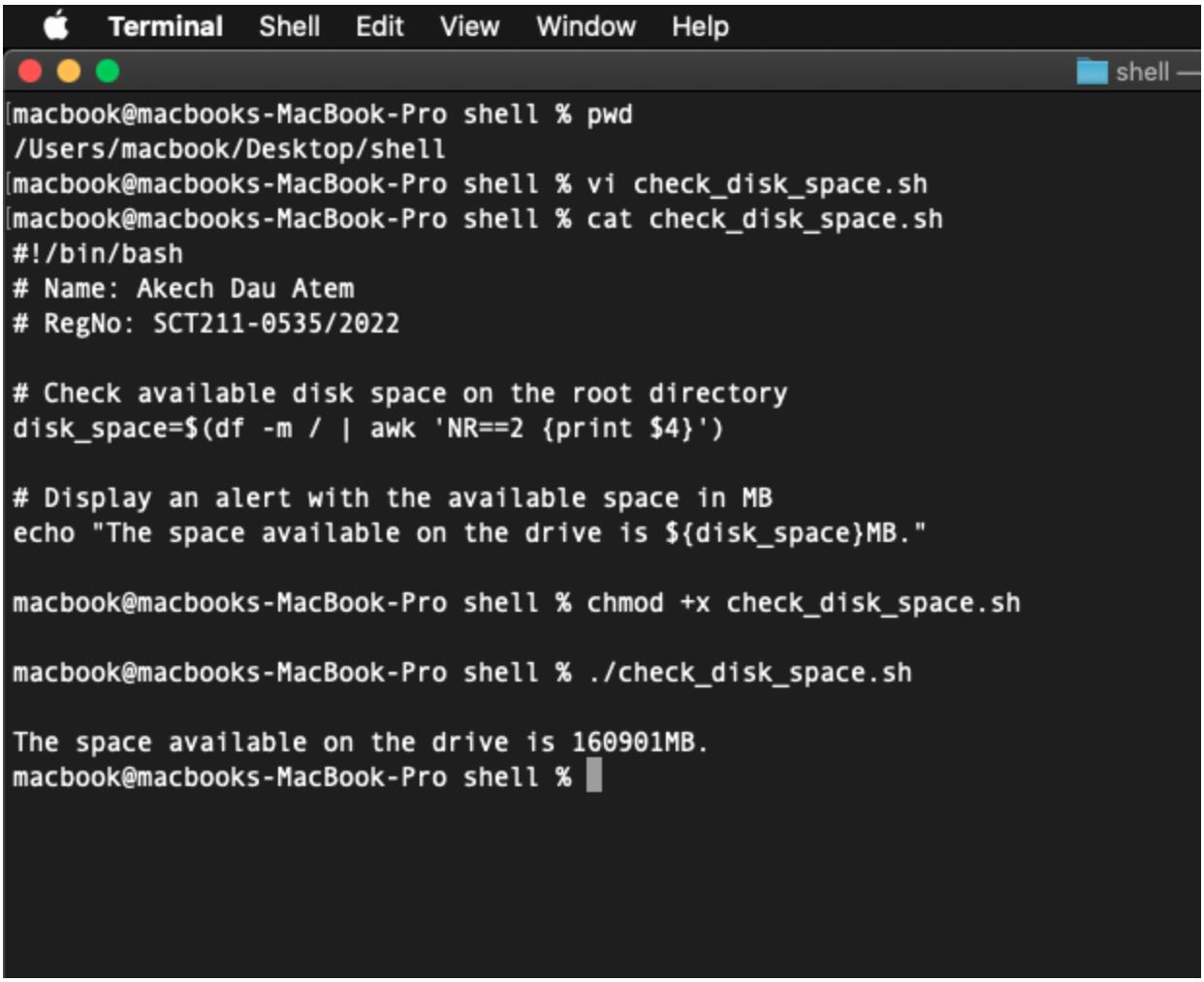
```
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022
```

```
# Prompt the user for the file containing the names and emails
read -p "Enter the file name containing the names and emails (e.g., students_list.txt): " file_name

# Check if the file exists
if [ ! -f "$file_name" ]; then
echo "Error: File '$file_name' does not exist."
exit 1
fi

# Read the file and send invitations
while IFS=',' read -r name email; do
echo "Sending invitation to $name <$email>..."
mail -s "Invitation to JhubAfrica Webinar" "$email" <<EOF
Dear $name,

You are invited to our 21st Open Webinar on Skills Optimization by JhubAfrica.
**Date:** Friday, 29th November
**Time:** 5:00 P.M.
**Zoom Link:** https://youtu.be/ONVTA7LKMIs

We will also broadcast the webinar on our YouTube channel.
**YouTube Link:** https://youtu.be/ONVTA7LKMIs

Looking forward to your participation.

Best regards,
Department of Computing
EOF
done < "$file_name"

echo "All invitations sent!"
```

Question 8

Task

Write a bash script that checks the available disk space on your drive (e.g., C:) and alerts the user with a message indicating the amount of space available in megabytes (MB).



Question 9

Task

Part (a): Using the wget command, download two files:

1. An example TXT file: https://github.com/yesinteractive/dadjokes/blob/master/controllers/jokes.txt
2. An example JSON file: https://gist.github.com/rominirani/8235702#file-employees-json

```
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# URLs to download
```

```
txt_url="https://raw.githubusercontent.com/yesinteractive/dadjokes/master/controllers/jokes.txt"
json_url="https://gist.githubusercontent.com/rominirani/8235702/raw/6820f70bd9187f9db51d97e089759fe92d8c1f1d/employees.json"

# Download the TXT file
wget -O jokes.txt "$txt_url"

# Download the JSON file
wget -O employees.json "$json_url"

# Verify if the files were downloaded
if [ -f "jokes.txt" ] && [ -f "employees.json" ]; then
echo "Files downloaded successfully."
else
echo "Error: Failed to download one or both files."
fi
```

Download Files:

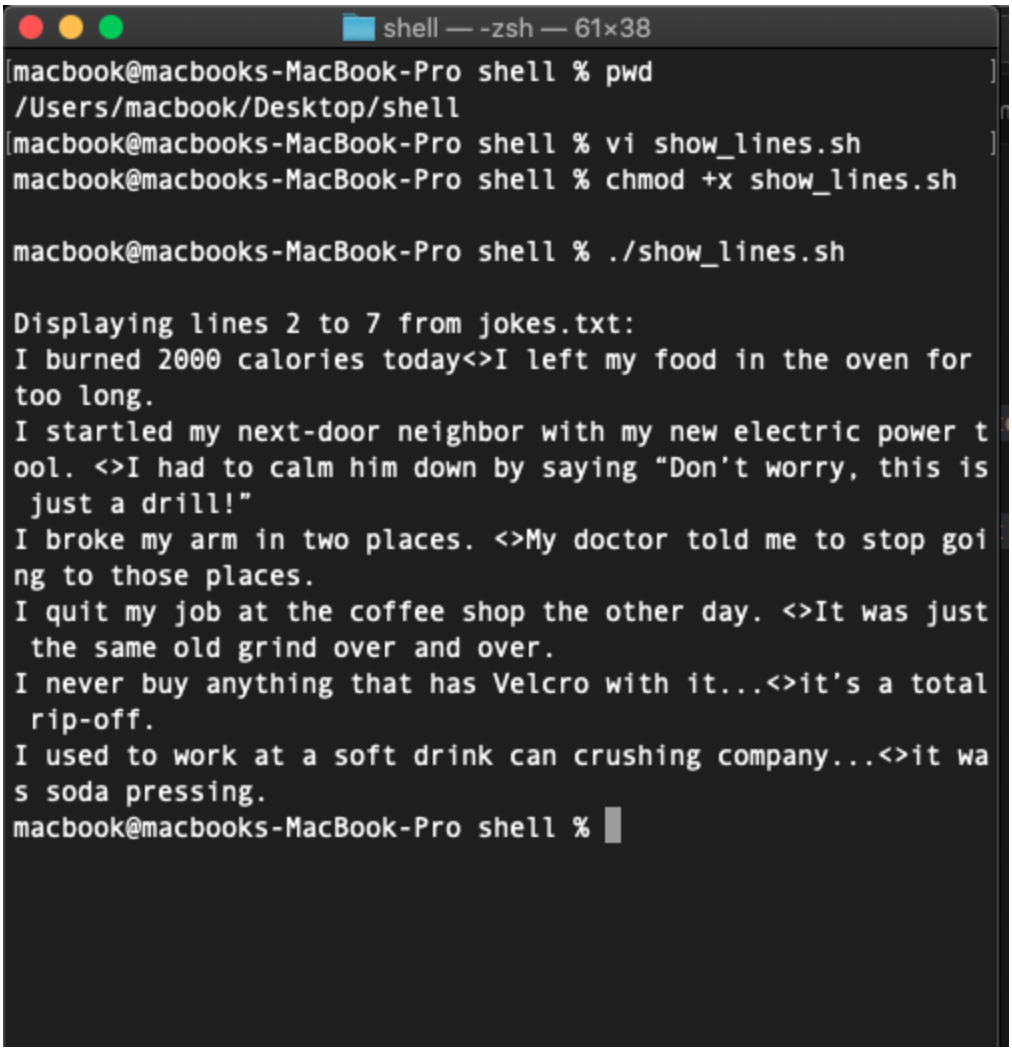
The wget -O command downloads the file and saves it with the specified name (jokes.txt and employees.json).

Verification:

The script checks if both files exist using [-f].

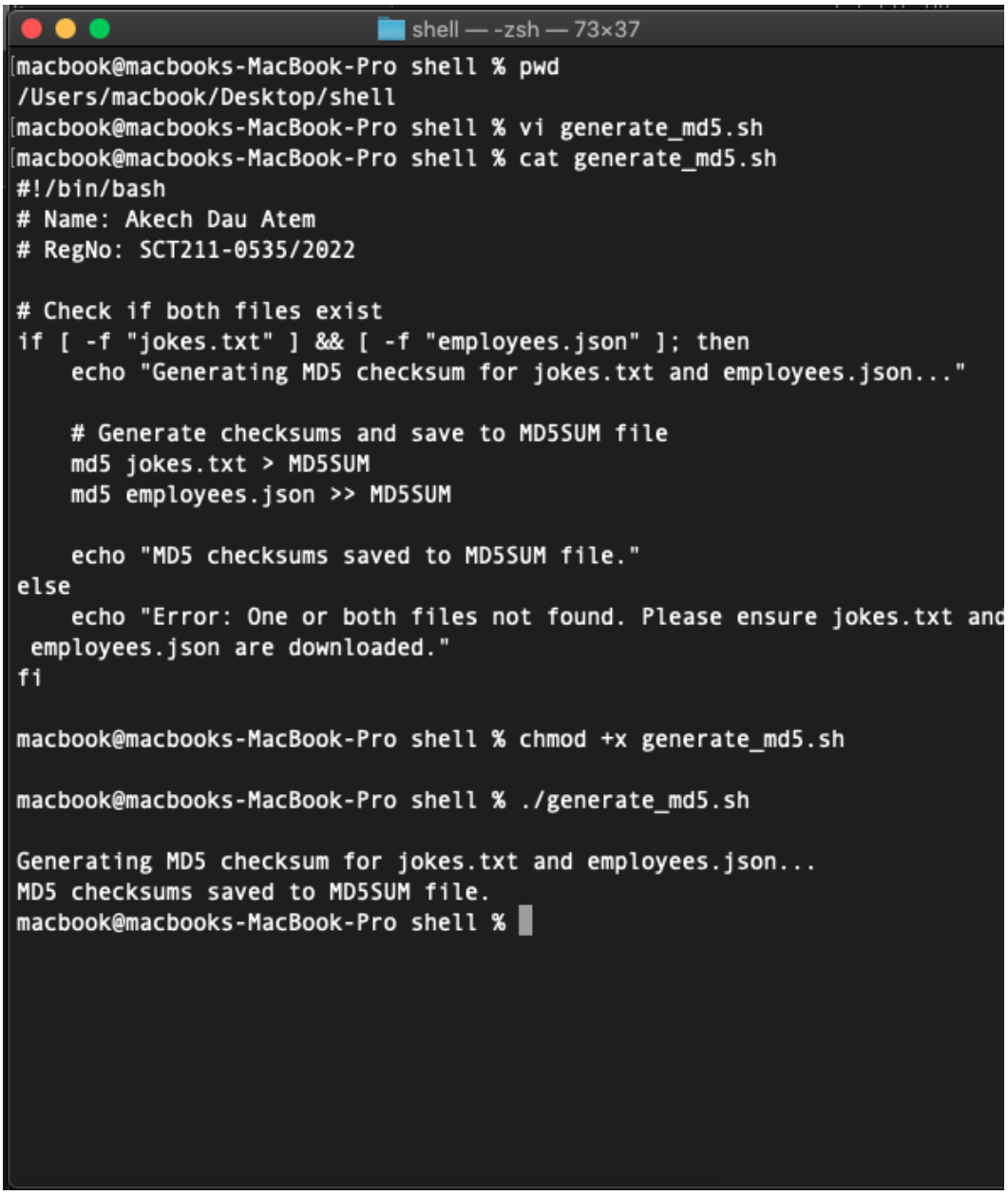
Part (b): Show Lines 2 to 7 of jokes.txt

Use the sed command to display lines 2 through 7:



Part (c): Create an MD5 Checksum for Both Files

This script generates an **MD5 checksum** for jokes.txt and employees.json, saving the checksums to a file named MD5SUM



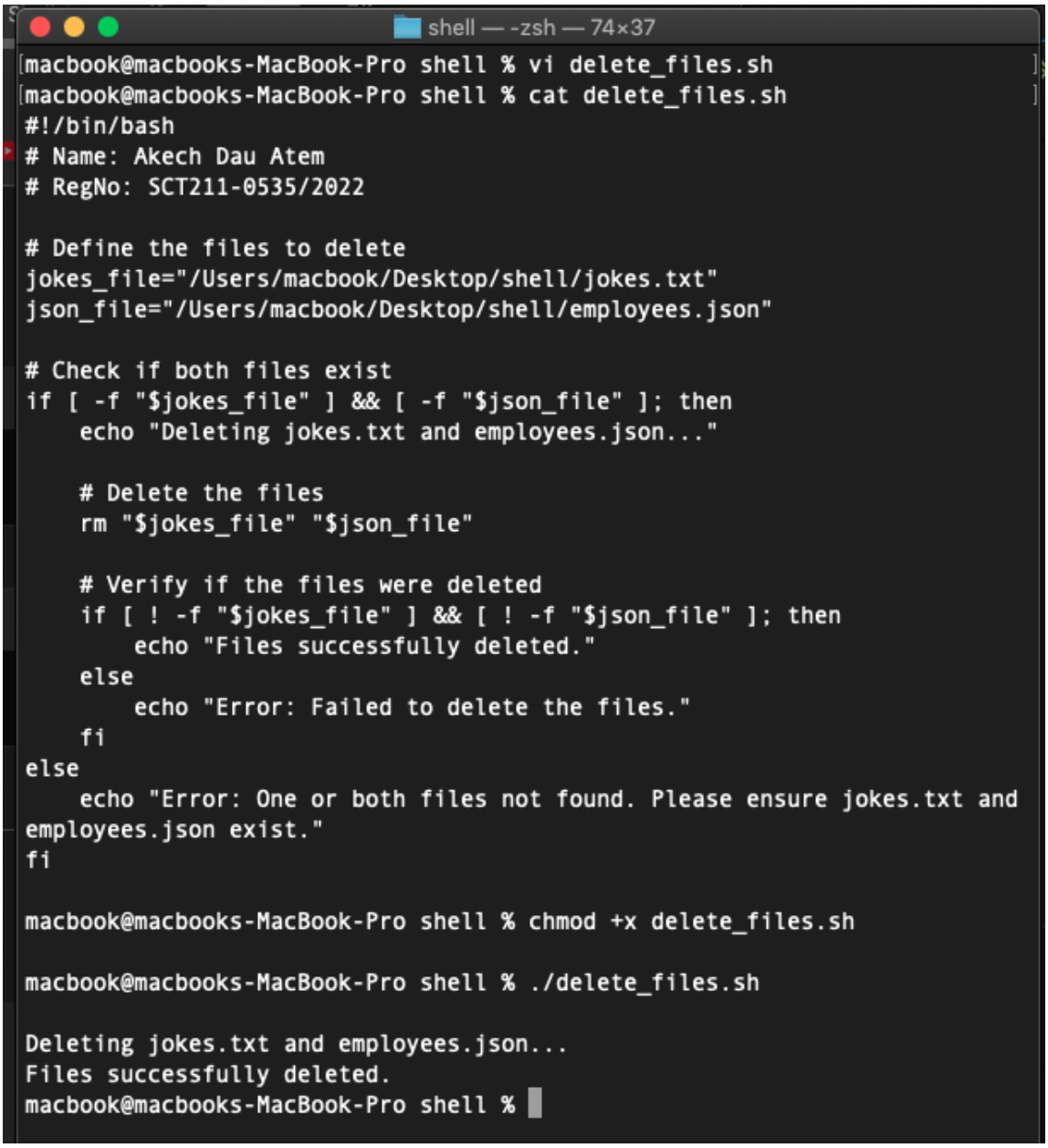
Part (d): Archive Both Files as Johnnie.tar.gz

Here's a script to create a tarball archive (Johnnie.tar.gz) containing both jokes.txt and employees.json.

```
macbook@macbooks-MacBook-Pro shell % vi archive_files.sh
macbook@macbooks-MacBook-Pro shell % cat archive_files.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Define the files to archive
jokes_file="/Users/macbook/Desktop/shell/jokes.txt"
json_file="/Users/macbook/Desktop/shell/employees.json"
archive_name="Johnnie.tar.gz"

# Check if both files exist
if [ -f "$jokes_file" ] && [ -f "$json_file" ]; then
    echo "Archiving jokes.txt and employees.json into $archive_name..."

    # Create the tarball archive
    tar -czf "$archive_name" -C "/Users/macbook/Desktop/shell" "jokes.txt" "employees.json"

    # Confirm the archive was created
    if [ -f "$archive_name" ]; then
        echo "Files successfully archived as $archive_name."
    else
        echo "Error: Failed to create the archive."
    fi
else
    echo "Error: One or both files not found. Please ensure jokes.txt and employees.json exist."
fi

macbook@macbooks-MacBook-Pro shell % chmod +x archive_files.sh

macbook@macbooks-MacBook-Pro shell % ./archive_files.sh

Archiving jokes.txt and employees.json into Johnnie.tar.gz...
Files successfully archived as Johnnie.tar.gz.
macbook@macbooks-MacBook-Pro shell %
```

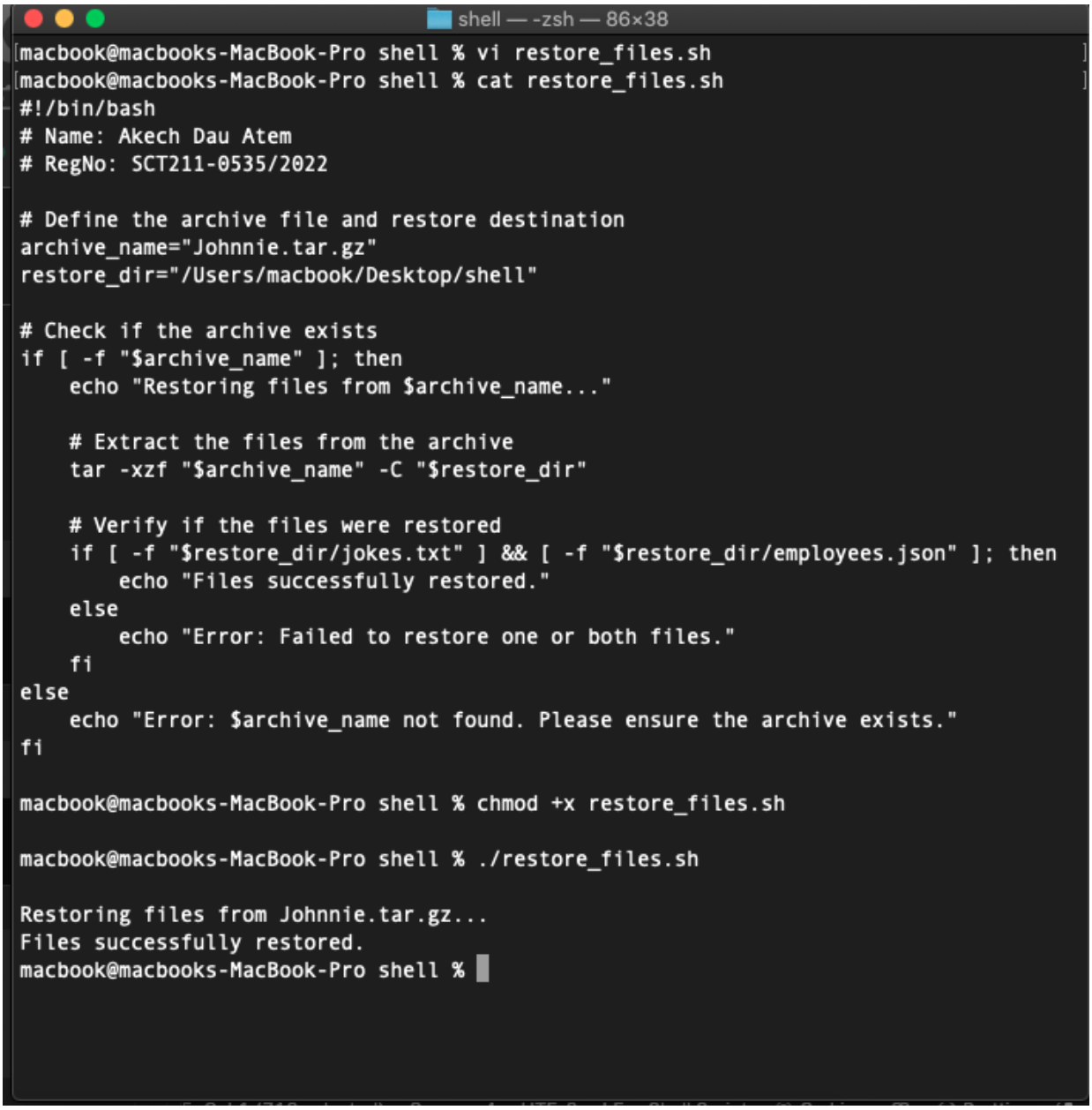
Part (e): Delete Both Files

This script will delete the downloaded files (jokes.txt and employees.json) after they have been archived.



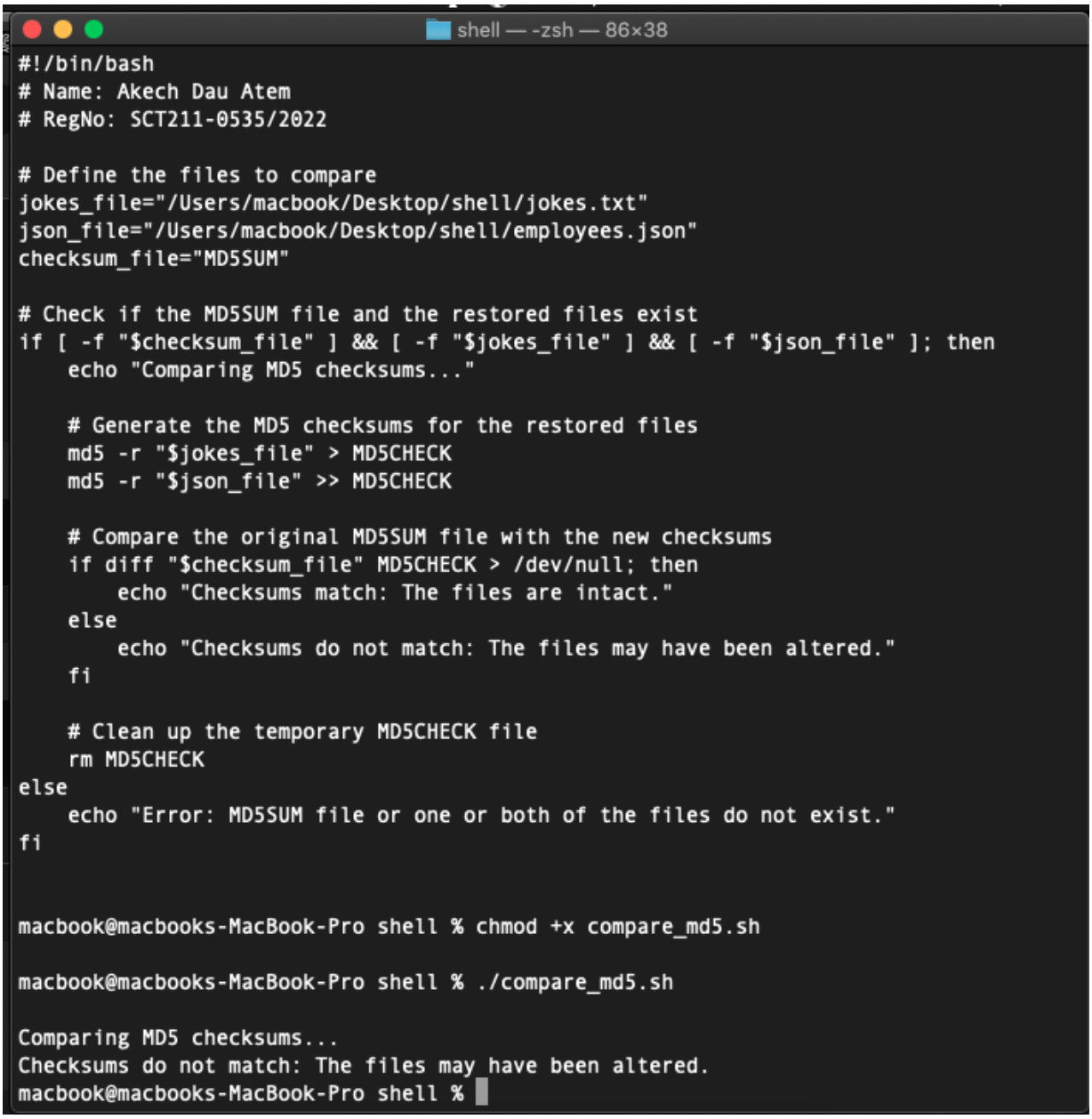
Part (f): Recover or Restore the Files from the Archive

This script will restore the files (jokes.txt and employees.json) from the Johnnie.tar.gz archive that was created in **Part (d)**.



Part (g): Compare MD5 Checksums

This script compares the MD5 checksums of the restored files (jokes.txt and empl


shell — -zsh — 86×38
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022
# Define the files to compare
jokes_file="/Users/macbook/Desktop/shell/jokes.txt"
json_file="/Users/macbook/Desktop/shell/employees.json"
checksum_file="MD5SUM"
# Check if the MD5SUM file and the restored files exist
if [-f "$checksum_file"] && [-f "$jokes_file"] && [-f "$json_file"]; then
echo "Comparing MD5 checksums..."
Generate the MD5 checksums for the restored files
md5 -r "$jokes_file" > MD5CHECK
md5 -r "$json_file" >> MD5CHECK
# Compare the original MD5SUM file with the new checksums
if diff "$checksum_file" MD5CHECK > /dev/null; then
echo "Checksums match: The files are intact."
else
echo "Checksums do not match: The files may have been altered."
fi
Clean up the temporary MD5CHECK file
rm MD5CHECK
else
echo "Error: MD5SUM file or one or both of the files do not exist."
fi
macbook@macbooks-MacBook-Pro shell % chmod +x compare_md5.sh
macbook@macbooks-MacBook-Pro shell % ./compare_md5.sh
Comparing MD5 checksums...
Checksums do not match: The files may have been altered.
macbook@macbooks-MacBook-Pro shell %