Task 1 – Basic Listing & Counting

Write a script that:

1. Takes **one argument**: a directory path.
2. Prints:
   - Total number of items in that directory.
   - Number of files.
   - Number of sub-directories.
3. If the directory does not exist, print a clear error message.

Example usage (for you to test):

./task1.sh /some/directory

Answer:

1. Opening **PowerShell**

2. Typed **wsl**

3. Used **cd ~** because it quickly takes you to your **real Linux home folder**, where your own Linux files and scripts actually belong.

4. Created new file by nano editor

nano task1.sh (as per given example)

```
d="$1"
[ -d "$d" ] || { echo "Directory does not exist!"; exit 1; }
total=$(ls -A "$d" | wc -l)
files=$(find "$d" -maxdepth 1 -type f | wc -l)
dirs=$(($(find "$d" -maxdepth 1 -type d | wc -l) - 1))
echo "Total No. of items in this directory: $total"
echo "No. of Files: $files"
echo "No. of Sub-Directories: $dirs"
```

NOTE:

CTRL + O → To Enter

CTRL +S → To Save

CTRL + X → To Exit

5.To Execute!

chmod +x task1.sh

6. Example as per given question:

./task1.sh /home/uday

RESULTxCASE-1(Exists Case):

Total No. of items in this directory: 19

No. of Files: 11

No. of Sub-Directories: 6

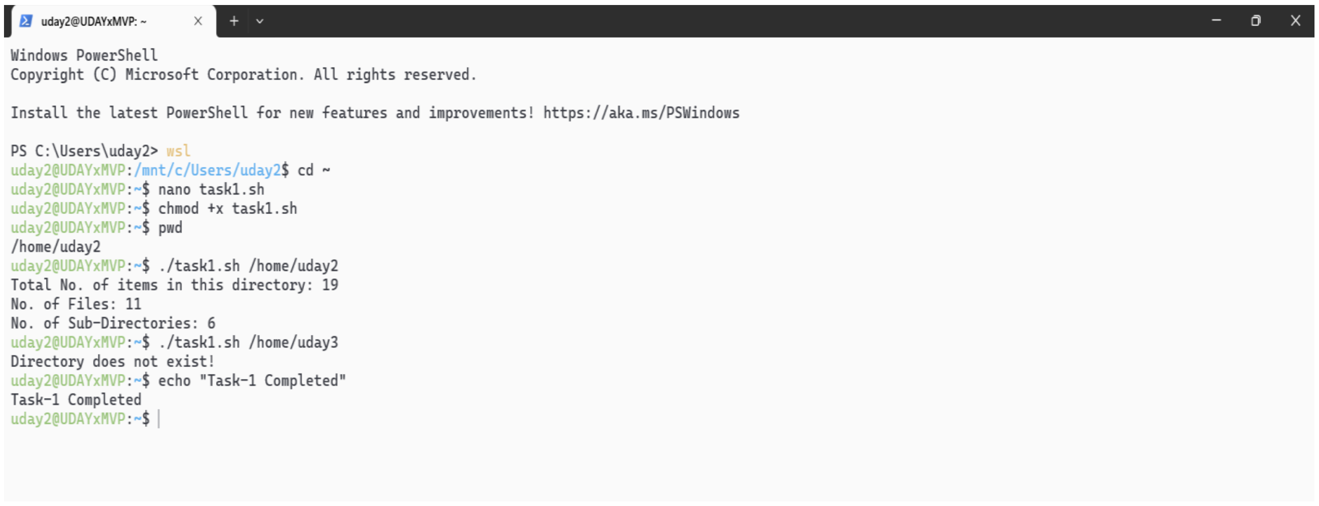
RESULTxCASE-2(Not Exists Case):

uday2@UDAYxMVP:~$ ./task1.sh /home/uday3

Directory does not exist!

uday2@UDAYxMVP:~$ echo "Task-1 Completed"

Task-1 Completed



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\uday2> wsl
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$ nano task1.sh
uday2@UDAYxMVP:~$ chmod +x task1.sh
uday2@UDAYxMVP:~$ pwd
/home/uday2
uday2@UDAYxMVP:~$ ./task1.sh /home/uday2
Total No. of items in this directory: 19
No. of Files: 11
No. of Sub-Directories: 6
uday2@UDAYxMVP:~$ ./task1.sh /home/uday3
Directory does not exist!
uday2@UDAYxMVP:~$ echo "Task-1 Completed"
Task-1 Completed
uday2@UDAYxMVP:~$ |
```

Task 2 – File Info Helper

Write a script that:

1. Asks the user to enter a **file name** (can be with or without path).
2. If the file exists, print:
   - File size in bytes.
   - Last modified time.
   - Permissions in symbolic form (e.g. `-rw-r--r--`).
3. If the file does not exist, print a suitable message.

Answer:

NOTE
If Vim isnt installed we can use:
sudo apt update
sudo apt install vim

NOTES:

```
i       → insert mode (start typing)
Esc     → go back to command mode
:w      → save (write) file
:q      → quit
:wq     → save and quit
:q!     → quit without saving
```

1. **cd ~**
2. **vim task2.sh**
3. press **i**

```
read -p "Enter file name or full path: " file

if [ ! -e "$file" ]; then
   echo "File does not exist!"
   exit 1
fi

size=$(stat -c%s "$file")
mtime=$(stat -c%y "$file")
perms=$(stat -c%A "$file")

echo "File:          $file"
echo "Size (bytes):  $size"
echo "Last modified: $mtime"
echo "Permissions:   $perms"
```

4. Write Code as per (as per question)
5. Esc
6. :wq
7. chmod +x task2.sh
8. echo "hello" > semster (Creating a file manually)
9. ./task2.sh
10. /home/uday2/semster

OUTPUT-1 (VALID NAME CASE):

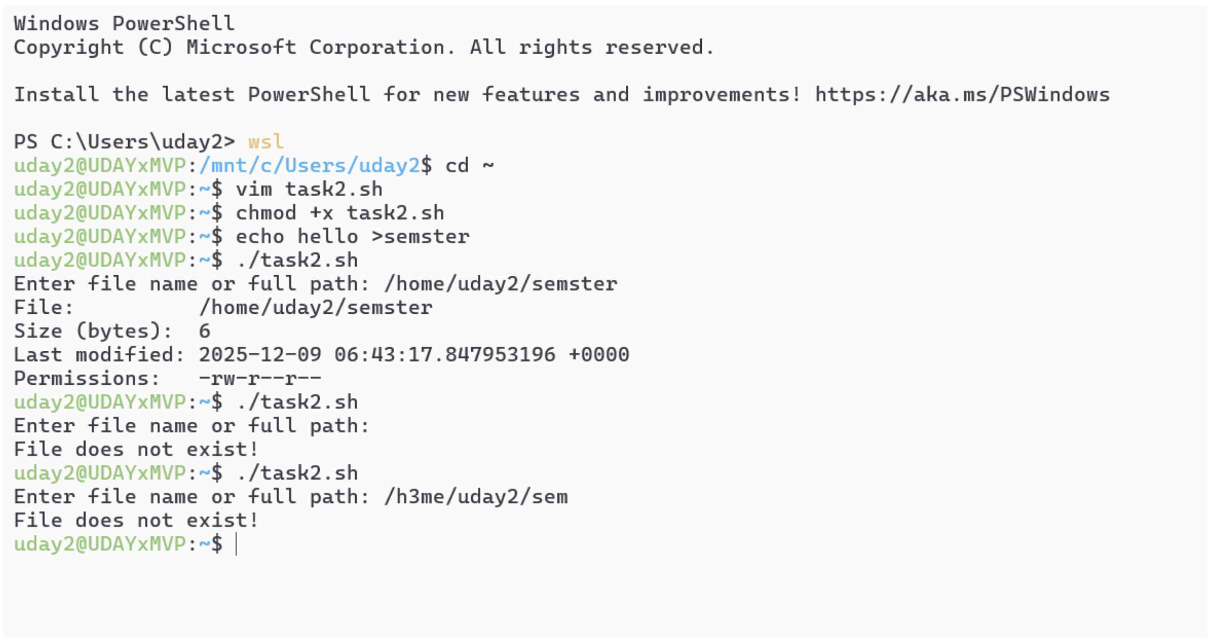
```
uday2@UDAYxMVP:~$ ./task2.sh
Enter file name or full path: /home/uday2/semster
File:          /home/uday2/semster
Size (bytes):  6
```

Last modified: 2025-12-09 06:43:17.847953196 +0000
Permissions: -rw-r--r—

& also handles invalid names and empty file names too!

OUTPUT-2 (EMPTY FILE CASE):
uday2@UDAYxMVP:~$ ./task2.sh
Enter file name or full path:
File does not exist!

OUTPUT-3 (WRONG FILE NAME CASE):
uday2@UDAYxMVP:~$ ./task2.sh
Enter file name or full path: /h3me/uday2/sem
File does not exist!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\uday2> wsl
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$ vim task2.sh
uday2@UDAYxMVP:~$ chmod +x task2.sh
uday2@UDAYxMVP:~$ echo hello >semster
uday2@UDAYxMVP:~$ ./task2.sh
Enter file name or full path: /home/uday2/semster
File:          /home/uday2/semster
Size (bytes):  6
Last modified: 2025-12-09 06:43:17.847953196 +0000
Permissions:   -rw-r--r--
uday2@UDAYxMVP:~$ ./task2.sh
Enter file name or full path:
File does not exist!
uday2@UDAYxMVP:~$ ./task2.sh
Enter file name or full path: /h3me/uday2/sem
File does not exist!
uday2@UDAYxMVP:~$ |
```

Notes:

Nano Editor:
A simple, beginner-friendly terminal text editor used mainly for quick file editing.

Vim Editor:
A powerful, mode-based terminal editor designed for speed and advanced editing.

Nano vs Vim

1. Learning Curve
Nano → Very easy for beginners.
Vim → Harder at first (mode-based editing).

2. Modes
Nano → Single mode (type directly).
Vim → Insert mode + Command mode.

3. Shortcuts
Nano → Uses visible shortcuts (Ctrl + O, Ctrl + X).
Vim → Uses key commands (:wq, dd, i, Esc).

4. Usage Purpose
Nano → Quick edits, small scripts, straightforward changes.
Vim → Fast coding, heavy editing, navigation without mouse.

5. Availability & Popularity
Nano → Common on most Linux systems, simpler.
Vim → Standard on all Unix-like systems, preferred by power users.

Task 3 – Text File Backup
Write a script that:

1. Takes **one argument**: a directory path.
2. Creates a new folder inside it named backup_YYYYMMDD (use the current date).
3. Copies all .txt files from the given directory into this backup folder.
4. Prints how many files were copied.

ANSWER:

1. wsl
2. cd ~
3. vim task3.sh

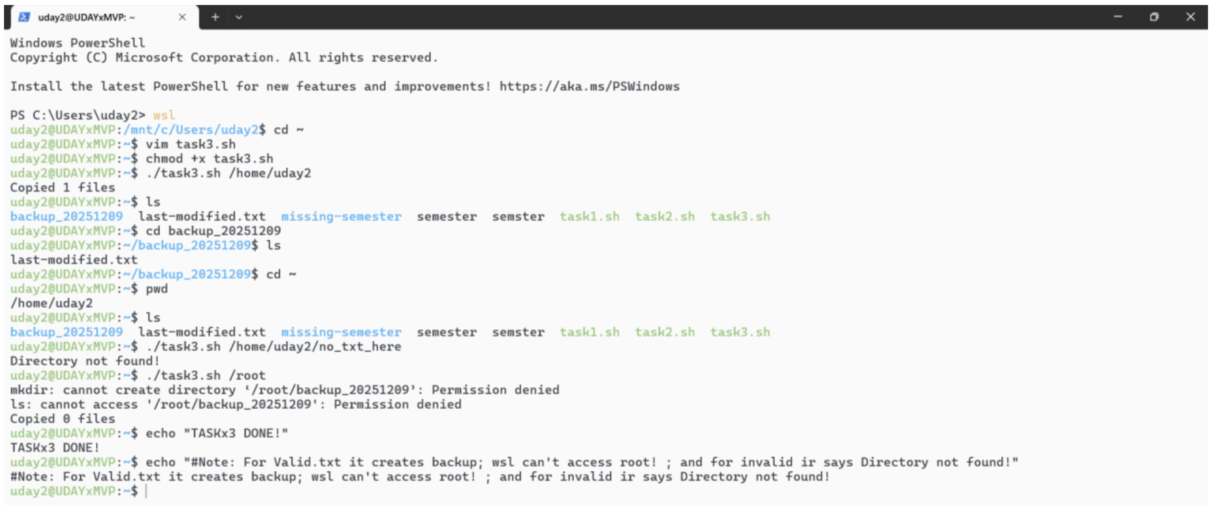
```
d="$1"
[ -d "$d" ] || { echo "Directory not found!"; exit 1; }

b="$d/backup_$(date +%Y%m%d)"
mkdir "$b"

cp "$d"/*.txt "$b" 2>/dev/null
echo "Copied $(ls "$b" | wc -l) files"
```

4. i→ esc → :wq
5. chmod +x task3.sh
6. ./task3.sh /home/uday2

OUTPUTx1: Valid.txt File Exists
If .txt files exist → they get copied.
A new folder gets created like:
backup_20250209

> **Copied 1 files**
> **uday2@UDAYxMVP:~$** ls
> **backup_20251209** last-modified.txt missing-semester semester semster task1.sh task2.sh task3.sh

(can be seen above new folder created)



uday2@UDAYxMVP: ~

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\uday2> wsl
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$ vim task3.sh
uday2@UDAYxMVP:~$ chmod +x task3.sh
uday2@UDAYxMVP:~$ ./task3.sh /home/uday2
Copied 1 files
uday2@UDAYxMVP:~$ ls
backup_20251209  last-modified.txt  missing-semester  semester  semster  task1.sh  task2.sh  task3.sh
uday2@UDAYxMVP:~$ cd backup_20251209
uday2@UDAYxMVP:~/backup_20251209$ ls
last-modified.txt
uday2@UDAYxMVP:~/backup_20251209$ cd ~
uday2@UDAYxMVP:~$ pwd
/home/uday2
uday2@UDAYxMVP:~$ ls
backup_20251209  last-modified.txt  missing-semester  semester  semster  task1.sh  task2.sh  task3.sh
uday2@UDAYxMVP:~$ ./task3.sh /home/uday2/no_txt_here
Directory not found!
uday2@UDAYxMVP:~$ ./task3.sh /root
mkdir: cannot create directory ‘/root/backup_20251209’: Permission denied
ls: cannot access '/root/backup_20251209': Permission denied
Copied 0 files
uday2@UDAYxMVP:~$ echo "TASKx3 DONE!"
TASKx3 DONE!
uday2@UDAYxMVP:~$ echo "#Note: For Valid.txt it creates backup; wsl can't access root! ; and for invalid ir says Directory not found!"
#Note: For Valid.txt it creates backup; wsl can't access root! ; and for invalid ir says Directory not found!
uday2@UDAYxMVP:~$ |
```

Task 4 – Top N Largest Files

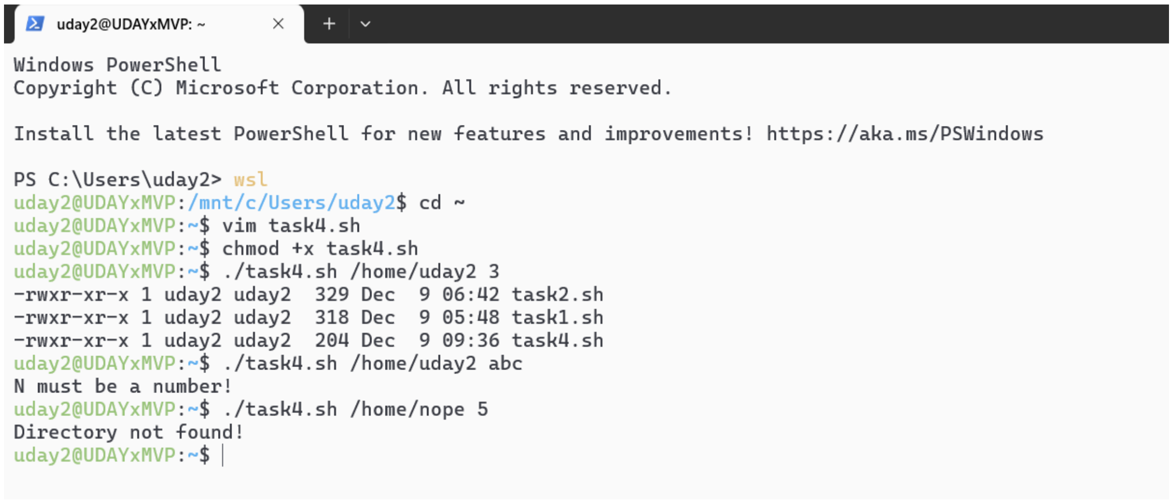
Write a script that:

1. Takes **two arguments**:
   - A directory path.
   - A number `N`.
2. Finds the **N largest regular files** inside that directory (non-recursive).
3. Prints their sizes and names in **descending** order of size.
4. If `N` is missing or not a number, handle the error.

Answer:

1) Open Windows PowerShell
2) Type: wsl
3) Go to home folder:
   cd ~
4) Open file in vim:
   vim task4.sh
5) Press 'i' and to write the script
6) Press Esc
7) Type :wq (save & exit)
8) Make it executable:
   chmod +x task4.sh
9) Run the script:
   ./task4.sh /home/uday2 3

```
d="$1"
n="$2"

if [ ! -d "$d" ]; then
   echo "Directory not found!"
   exit 1
fi

if ! [[ "$n" =~ ^[0-9]+$ ]]; then
   echo "N must be a number!"
   exit 1
fi

ls -lS "$d" | grep "^-" | head -n "$n"
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\uday2> wsl
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$ vim task4.sh
uday2@UDAYxMVP:~$ chmod +x task4.sh
uday2@UDAYxMVP:~$ ./task4.sh /home/uday2 3
-rwxr-xr-x 1 uday2 uday2  329 Dec  9 06:42 task2.sh
-rwxr-xr-x 1 uday2 uday2  318 Dec  9 05:48 task1.sh
-rwxr-xr-x 1 uday2 uday2  204 Dec  9 09:36 task4.sh
uday2@UDAYxMVP:~$ ./task4.sh /home/uday2 abc
N must be a number!
uday2@UDAYxMVP:~$ ./task4.sh /home/nope 5
Directory not found!
uday2@UDAYxMVP:~$
```

OUTPUT COVERING VALID AND INVALID CASES CAN BE SEEN ABOVE!

Task 5 – Simple Log Search

Write a script that:

1. Takes **two arguments**:
    - A directory path.

    - A search word.

2. Searches all `.log` files inside that directory (non-recursive) for lines that contain the word.

3. For every `.log` file that has at least one match, print the file name and number of matching lines.

If no `.log` files are found, print a message

ANSWER:

1) Open Windows PowerShell
2) Type: wsl
3) Go to home folder:
   cd ~
4) Open script in vim:
   vim task5.sh
5) Press i → to write code
6) Esc
7) :wq
8) Make executable:
   chmod +x task5.sh
9) Run:
   ./task5.sh /home/uday2 error

```
d="$1"
word="$2"

if [! -d "$d"]; then
   echo "Directory not found!"
   exit 1
fi

logs=$(ls "$d"/*.log 2>/dev/null)
[ -z "$logs"] && {echo "No .log files found!"; exit 0;}

for f in $logs; do
   count=$(grep -c "$word" "$f")
   [ "$count" -gt 0] && echo "$f: $count matches"
done
```

NOTES:

d="$1"
→ First argument = directory.

word="$2"
→ Second argument = the word to search.

if [! -d "$d"]
→ Checks if directory is missing.

logs=$(ls "$d"/*.log 2>/dev/null)
→ Saves all .log files into 'logs'. Hides errors.

[-z "$logs"]
→ If empty, means no .log files exist.

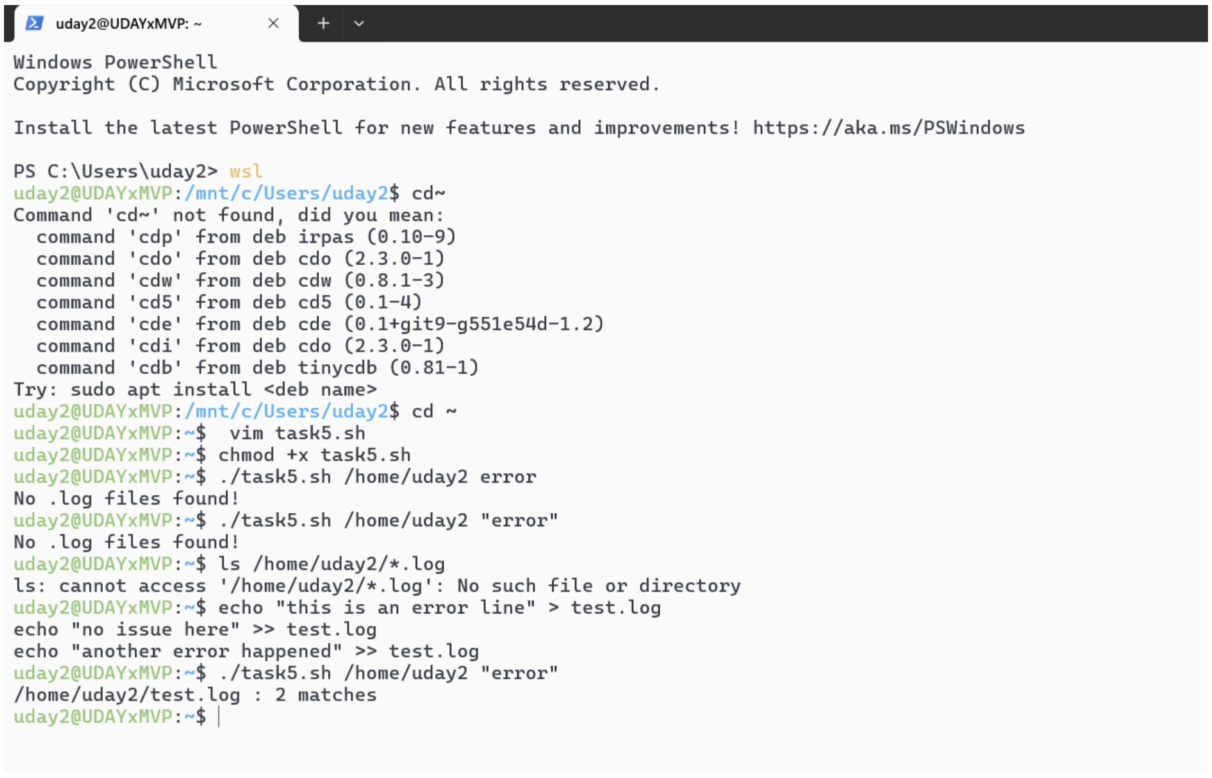
for f in $logs
→ Loop through each log file.

count=$(grep -c "$word" "$f")
→ Count how many lines contain the word.

["$count" -gt 0] && echo "$f: $count matches"
→ Print only if matches exist.

OUTPUT:

```
uday2@UDAYxMVP:~$ ./task5.sh /home/uday2 "error"
/home/uday2/test.log :2 matches
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\uday2> wsl
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd~
Command 'cd~' not found, did you mean:
  command 'cdp' from deb irpas (0.10-9)
  command 'cdo' from deb cdo (2.3.0-1)
  command 'cdw' from deb cdw (0.8.1-3)
  command 'cd5' from deb cd5 (0.1-4)
  command 'cde' from deb cde (0.1+git9-g551e54d-1.2)
  command 'cdi' from deb cdo (2.3.0-1)
  command 'cdb' from deb tinycdb (0.81-1)
Try: sudo apt install <deb name>
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$  vim task5.sh
uday2@UDAYxMVP:~$ chmod +x task5.sh
uday2@UDAYxMVP:~$ ./task5.sh /home/uday2 error
No .log files found!
uday2@UDAYxMVP:~$ ./task5.sh /home/uday2 "error"
No .log files found!
uday2@UDAYxMVP:~$ ls /home/uday2/*.log
ls: cannot access '/home/uday2/*.log': No such file or directory
uday2@UDAYxMVP:~$ echo "this is an error line" > test.log
echo "no issue here" >> test.log
echo "another error happened" >> test.log
uday2@UDAYxMVP:~$ ./task5.sh /home/uday2 "error"
/home/uday2/test.log : 2 matches
uday2@UDAYxMVP:~$ |
```

Task 6 – Hidden Files Report

Write a script that:

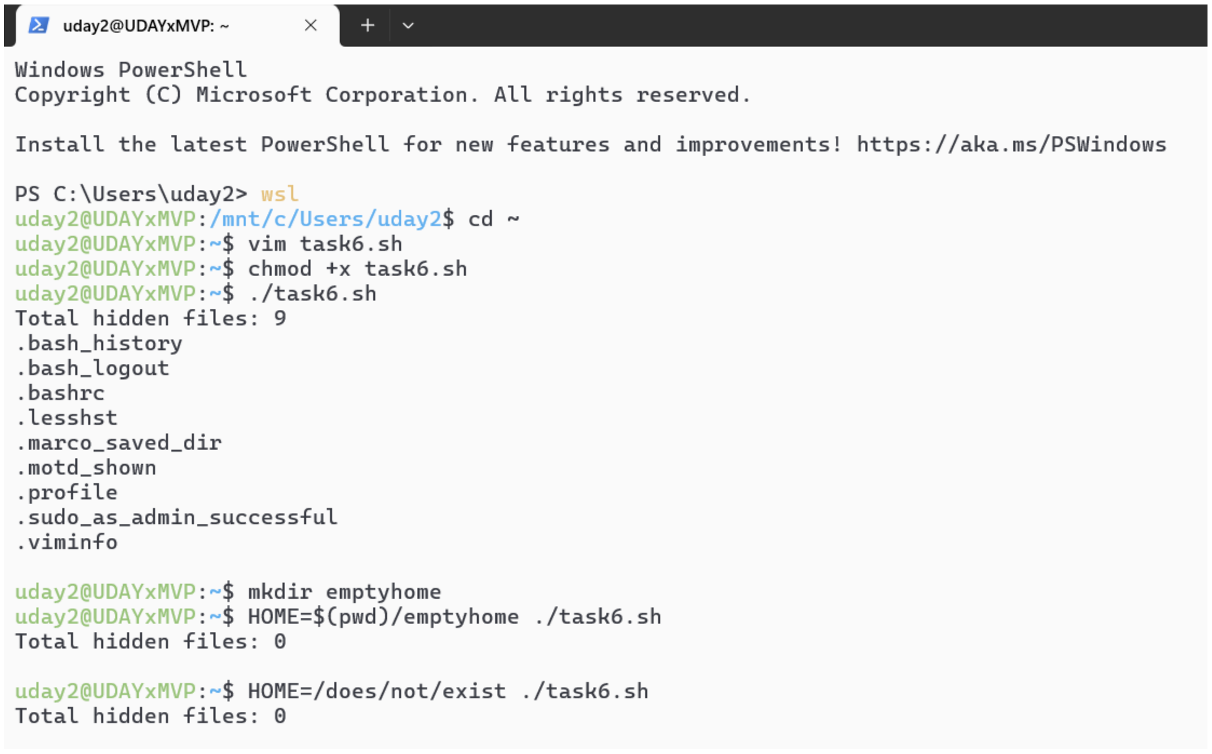
1. Looks in the **home directory** of the current user.
2. Lists all **hidden files** (not directories) directly inside it (non-recursive).
3. Prints:
   - Total number of hidden files.
   - A list of their names.

ANSWER:

1) Open Windows PowerShell
2) Type: wsl
3) Go to home:
   cd ~
4) Open script:
   vim task6.sh
5) Press i → to write code

6) Press Esc → :wq
7) Make executable:
   chmod +x task6.sh
8) Run:
   ./task6.sh

```
home="$HOME"

files=$(ls -A "$home"/.* 2>/dev/null | grep -v "/\.$" | grep -v "/\..\$" | grep -v
"/\.$")

count=0
list=""

for f in $files; do
   [ -f "$f"] && {count=$((count+1)); list="$list$(basename "$f")\n";}
done

echo "Total hidden files: $count"
echo -e "$list"
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\uday2> wsl
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$ vim task6.sh
uday2@UDAYxMVP:~$ chmod +x task6.sh
uday2@UDAYxMVP:~$ ./task6.sh
Total hidden files: 9
.bash_history
.bash_logout
.bashrc
.lesshst
.marco_saved_dir
.motd_shown
.profile
.sudo_as_admin_successful
.viminfo

uday2@UDAYxMVP:~$ mkdir emptyhome
uday2@UDAYxMVP:~$ HOME=$(pwd)/emptyhome ./task6.sh
Total hidden files: 0

uday2@UDAYxMVP:~$ HOME=/does/not/exist ./task6.sh
Total hidden files: 0
```

Test Cases

Test 1 – Normal case: Home has hidden files
./task6.sh
Output example:
Total hidden files: 9
.bashrc
.profile
.gitconfig (as shown in figure)
(PASS)

Test 2 – Home has zero hidden files
(rare on Linux, but possible if cleaned)
Create a test folder:
mkdir emptyhome
HOME=$(pwd)/emptyhome ./task6.sh
Output:
Total hidden files: 0
(PASS)

Test 3 – Fake/invalid HOME path
HOME=/does/not/exist ./task6.sh
Output:
Total hidden files: 0
(Edge case handled automatically)

Task 7 – Simple Change Tracker

Write a script that:

1. Takes **one argument**: a directory path.
2. Creates (or updates) a file named `.snapshot.txt` inside that directory containing the list of files (non-recursive, one per line).
3. On the **next run**, compares the current list of files with `.snapshot.txt` and prints:
   - Files that are **new**.
   - Files that are **missing** compared to the last run.
4. After printing, update `.snapshot.txt` with the new list.

ANSWER:

1) Open PowerShell

2) Type: wsl

3) Move to home:

```
cd ~
```

4) Create script:

```
vim task7.sh
```

5) Press i and to write the code which is below

6) Press Esc

7) Type :wq (save and quit)

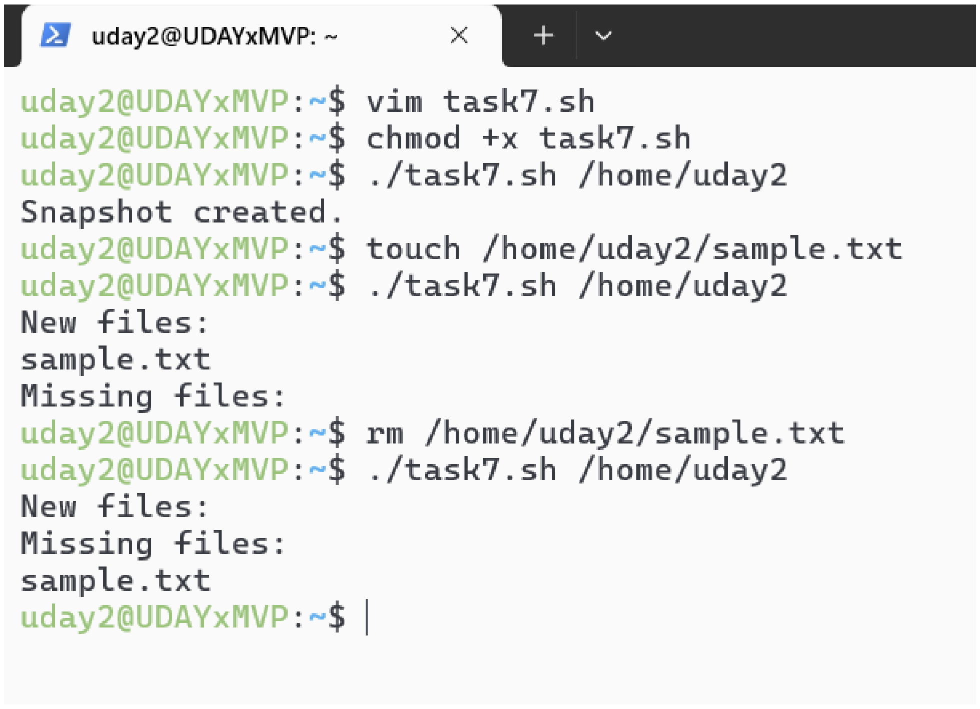
8) Make it executable:

```
chmod +x task7.sh
```

9) Run:

```
./task7.sh /home/uday2
```

```
dir="$1"
snap="$dir/.snapshot.txt"

[ -d "$dir"] || {echo "Directory not found!"; exit 1;}

ls "$dir" | sort > /tmp/newlist.txt

# First run: no snapshot file exists
if [ ! -f "$snap"]; then
    cp /tmp/newlist.txt "$snap"
    echo "Snapshot created."
    exit 0
fi

echo "New files:"
comm -13 "$snap" /tmp/newlist.txt

echo "Missing files:"
comm -23 "$snap" /tmp/newlist.txt

# Update snapshot
cp /tmp/newlist.txt "$snap"
```



```
uday2@UDAYxMVP:~$ vim task7.sh
uday2@UDAYxMVP:~$ chmod +x task7.sh
uday2@UDAYxMVP:~$ ./task7.sh /home/uday2
Snapshot created.
uday2@UDAYxMVP:~$ touch /home/uday2/sample.txt
uday2@UDAYxMVP:~$ ./task7.sh /home/uday2
New files:
sample.txt
Missing files:
uday2@UDAYxMVP:~$ rm /home/uday2/sample.txt
uday2@UDAYxMVP:~$ ./task7.sh /home/uday2
New files:
Missing files:
sample.txt
uday2@UDAYxMVP:~$
```

Task 8 – Basic Symlink Checker

Write a script that:

1. Takes **one argument**: a directory path.
2. Lists all **symbolic links** directly inside that directory.
3. For each symbolic link, prints whether it points to a valid target or a missing target.

ANSWER:

1) Open Windows PowerShell

2) Type: wsl

3) Go to home:

```
cd ~
```

4) Open script:

```
vim task8.sh
```

5) Press i → write code

6) Press Esc → :wq

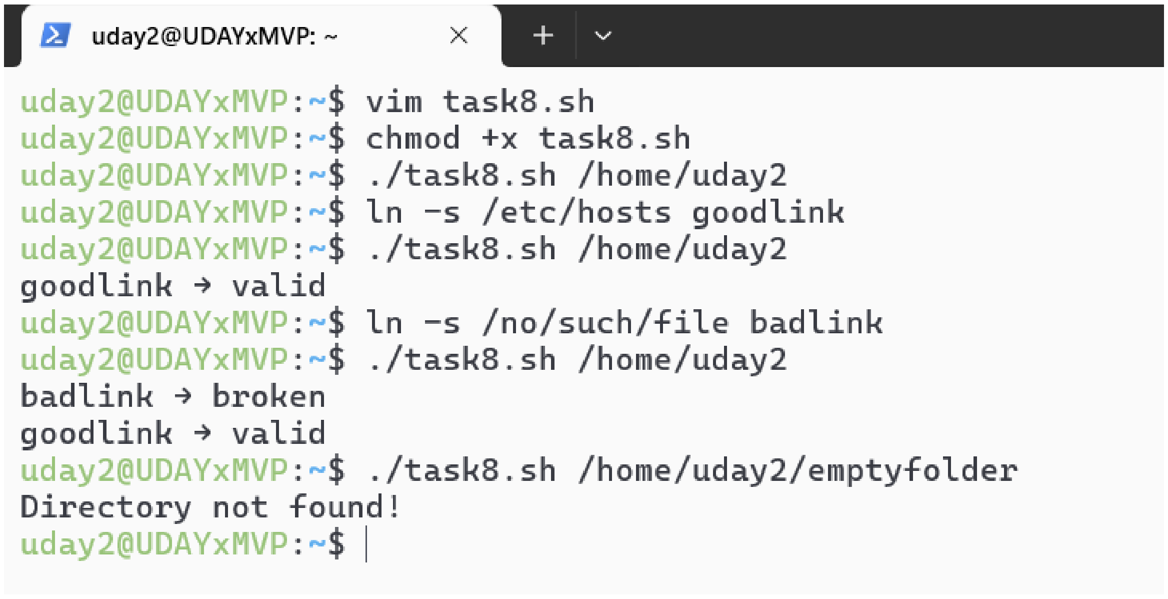
7) Make executable:

```
chmod +x task8.sh
```

8) Run:

```
./task8.sh /home/uday2
```

```
dir="$1"

[ -d "$dir"] || {echo "Directory not found!"; exit 1;}

for f in "$dir"/*; do
    [ -L "$f"] || continue
    if [ -e "$f"]; then
        echo "$(basename "$f") → valid"
    else
        echo "$(basename "$f") → broken"
    fi
done
```



uday2@UDAYxMVP: ~

```
uday2@UDAYxMVP:~$ vim task8.sh
uday2@UDAYxMVP:~$ chmod +x task8.sh
uday2@UDAYxMVP:~$ ./task8.sh /home/uday2
uday2@UDAYxMVP:~$ ln -s /etc/hosts goodlink
uday2@UDAYxMVP:~$ ./task8.sh /home/uday2
goodlink → valid
uday2@UDAYxMVP:~$ ln -s /no/such/file badlink
uday2@UDAYxMVP:~$ ./task8.sh /home/uday2
badlink → broken
goodlink → valid
uday2@UDAYxMVP:~$ ./task8.sh /home/uday2/emptyfolder
Directory not found!
uday2@UDAYxMVP:~$ 
```

Task 9 – Space Threshold Action

Write a script that:

1. Takes **one argument**: a directory path.
2. Calculates the total size (in MB) of that directory (non-recursive).
3. If the size is greater than a fixed threshold you choose (for example, 100 MB):
   - Print a warning message.
   - Create a file named `size_warning.txt` inside that directory containing the date and the size.

If the size is less than or equal to the threshold, print that everything is OK.

ANSWER:

1) Open Windows PowerShell

2) Type: wsl

3) Go home:

```
cd ~
```

4) Open script:

    vim task9.sh

5) Press i → write script

6) Esc → :wq

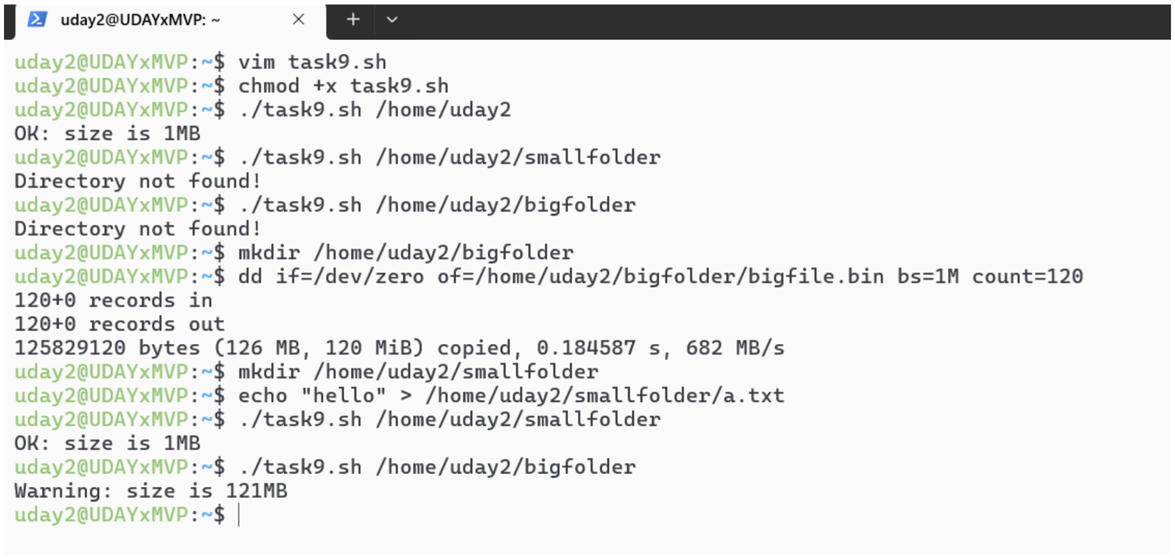
7) Make executable:

    chmod +x task9.sh

8) Run:

    ./task9.sh /home/uday2

```
dir="$1"
[ -d "$dir"] || {echo "Directory not found!"; exit 1;}

size=$(du -sm "$dir" | cut -f1)
th=100

if [ "$size" -gt "$th"]; then
   echo "Warning: size is ${size}MB"
   date +"%Y-%m-%d %H:%M Size: ${size}MB" > "$dir/size_warning.txt"
else
   echo "OK: size is ${size}MB"
fi
```



```
uday2@UDAYxMVP:~$ vim task9.sh
uday2@UDAYxMVP:~$ chmod +x task9.sh
uday2@UDAYxMVP:~$ ./task9.sh /home/uday2
OK: size is 1MB
uday2@UDAYxMVP:~$ ./task9.sh /home/uday2/smallfolder
Directory not found!
uday2@UDAYxMVP:~$ ./task9.sh /home/uday2/bigfolder
Directory not found!
uday2@UDAYxMVP:~$ mkdir /home/uday2/bigfolder
uday2@UDAYxMVP:~$ dd if=/dev/zero of=/home/uday2/bigfolder/bigfile.bin bs=1M count=120
120+0 records in
120+0 records out
125829120 bytes (126 MB, 120 MiB) copied, 0.184587 s, 682 MB/s
uday2@UDAYxMVP:~$ mkdir /home/uday2/smallfolder
uday2@UDAYxMVP:~$ echo "hello" > /home/uday2/smallfolder/a.txt
uday2@UDAYxMVP:~$ ./task9.sh /home/uday2/smallfolder
OK: size is 1MB
uday2@UDAYxMVP:~$ ./task9.sh /home/uday2/bigfolder
Warning: size is 121MB
uday2@UDAYxMVP:~$
```

Task 10 – Simple File Type Classifier (By Extension)

Write a script that:

1. Takes **one argument**: a directory path.
2. For each item in that directory (non-recursive):
   - If it is a file, check its extension (e.g. `.txt`, `.sh`, `.jpg`).
   - Count how many files there are of each extension.
3. At the end, print a summary like:
   - `txt: 5 files`
   - `sh: 3 files`
   - `no extension: 2 files`

1) Open PowerShell

2) Type: wsl

3) Go to home:

```
cd ~
```

4) Make script:

```
vim task10.sh
```

5) Press i → write code

6) Press Esc → :wq

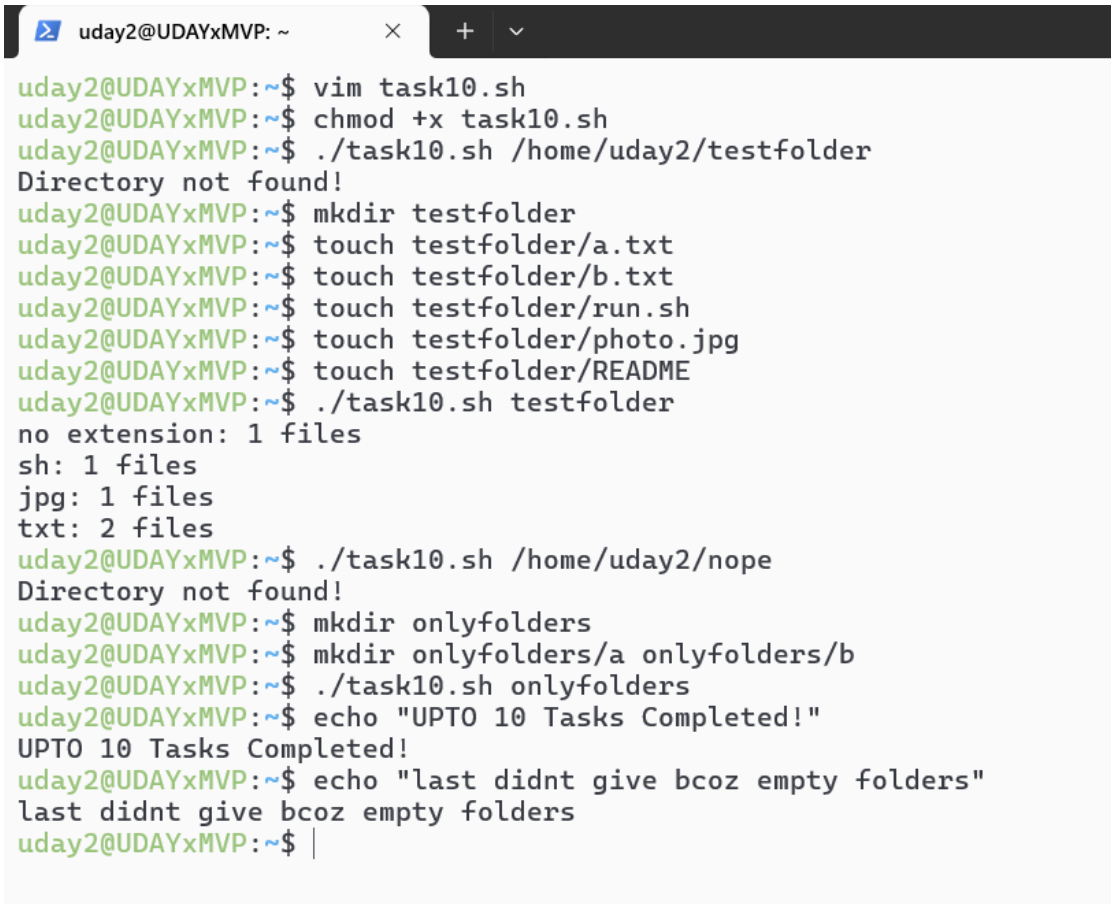
7) Make executable:

```
chmod +x task10.sh
```

8) Run:

```
./task10.sh /home/uday2/testfolder
```

```
dir="$1"
[ -d "$dir"] || {echo "Directory not found!"; exit 1;}

declare -A count

for f in "$dir"/*; do
   [ -f "$f"] || continue
   ext="${f##*.}"
   [ "$ext" = "$f"] && ext="no extension"
   count["$ext"]=$((${count["$ext"]} + 1))
done

for e in "${!count[@]}"; do
   echo "$e: ${count[$e]} files"
done
```



uday2@UDAYxMVP: ~

```
uday2@UDAYxMVP:~$ vim task10.sh
uday2@UDAYxMVP:~$ chmod +x task10.sh
uday2@UDAYxMVP:~$ ./task10.sh /home/uday2/testfolder
Directory not found!
uday2@UDAYxMVP:~$ mkdir testfolder
uday2@UDAYxMVP:~$ touch testfolder/a.txt
uday2@UDAYxMVP:~$ touch testfolder/b.txt
uday2@UDAYxMVP:~$ touch testfolder/run.sh
uday2@UDAYxMVP:~$ touch testfolder/photo.jpg
uday2@UDAYxMVP:~$ touch testfolder/README
uday2@UDAYxMVP:~$ ./task10.sh testfolder
no extension: 1 files
sh: 1 files
jpg: 1 files
txt: 2 files
uday2@UDAYxMVP:~$ ./task10.sh /home/uday2/nope
Directory not found!
uday2@UDAYxMVP:~$ mkdir onlyfolders
uday2@UDAYxMVP:~$ mkdir onlyfolders/a onlyfolders/b
uday2@UDAYxMVP:~$ ./task10.sh onlyfolders
uday2@UDAYxMVP:~$ echo "UPTO 10 Tasks Completed!"
UPTO 10 Tasks Completed!
uday2@UDAYxMVP:~$ echo "last didnt give bcoz empty folders"
last didnt give bcoz empty folders
uday2@UDAYxMVP:~$ 
```

Tab 2

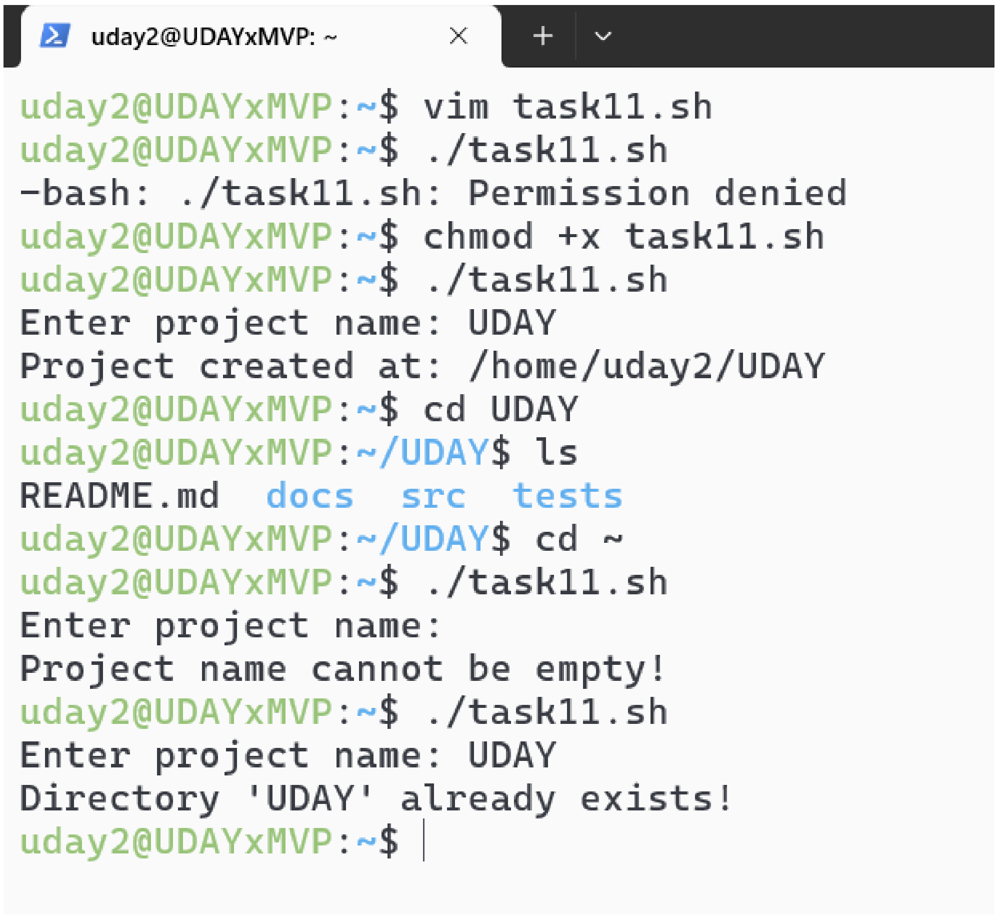
Task 11 – Project Workspace Initializer (Create)

Write a script that:

1. Asks the user for a **project name**.
2. Creates a directory with that name in the current directory.
3. Inside it, creates the following structure:
   - `src/`
   - `docs/`
   - `tests/`
4. Creates an empty file `README.md` inside the project directory and writes the project name as the first line.
5. Prints the full path of the created project directory.

ANSWER:

```
echo -n "Enter project name: "
read project

if [ -z "$project" ]; then
  echo "Project name cannot be empty!"
  exit 1
fi

if [ -d "$project" ]; then
  echo "Directory '$project' already exists!"
  exit 1
fi

mkdir -p "$project"/{src,docs,tests}

echo "$project" > "$project/README.md"

echo "Project created at: $(pwd)/$project"
```



```
uday2@UDAYxMVP:~$ vim task11.sh
uday2@UDAYxMVP:~$ ./task11.sh
-bash: ./task11.sh: Permission denied
uday2@UDAYxMVP:~$ chmod +x task11.sh
uday2@UDAYxMVP:~$ ./task11.sh
Enter project name: UDAY
Project created at: /home/uday2/UDAY
uday2@UDAYxMVP:~$ cd UDAY
uday2@UDAYxMVP:~/UDAY$ ls
README.md  docs  src  tests
uday2@UDAYxMVP:~/UDAY$ cd ~
uday2@UDAYxMVP:~$ ./task11.sh
Enter project name:
Project name cannot be empty!
uday2@UDAYxMVP:~$ ./task11.sh
Enter project name: UDAY
Directory 'UDAY' already exists!
uday2@UDAYxMVP:~$ 
```

Task 12 – Directory Content Viewer (Read)

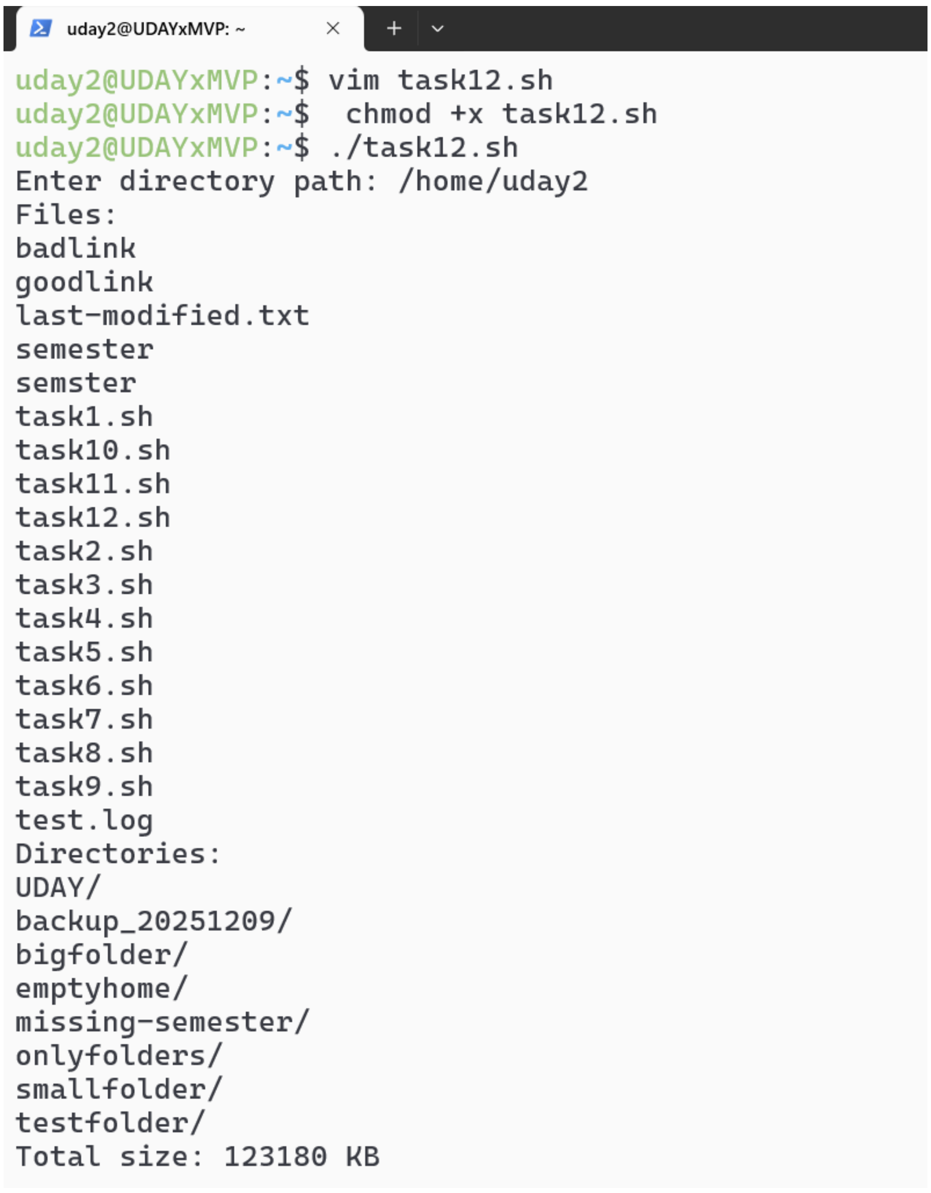
Write a script that:

1. Asks the user to enter a directory path.
2. If the directory exists, print:
   - A list of all files.
   - A list of all sub-directories.
3. Show the **total size** (in KB) of all files directly inside that directory (non-recursive).
4. If the directory does not exist, print an error message.

You may use commands like `ls`, `du`, `find` with options, etc.

ANSWER:

```
echo -n "Enter directory path: "
read dir

[ -d "$dir"] || {echo "Directory not found!"; exit 1;}

echo "Files:"
ls -p "$dir" | grep -v / || echo "No files"

echo "Directories:"
ls -p "$dir" | grep / || echo "No directories"

size=$(du -sk "$dir" | cut -f1)
echo "Total size: ${size} KB"
```



```
uday2@UDAYxMVP:~$ vim task12.sh
uday2@UDAYxMVP:~$  chmod +x task12.sh
uday2@UDAYxMVP:~$ ./task12.sh
Enter directory path: /home/uday2
Files:
badlink
goodlink
last-modified.txt
semester
semster
task1.sh
task10.sh
task11.sh
task12.sh
task2.sh
task3.sh
task4.sh
task5.sh
task6.sh
task7.sh
task8.sh
task9.sh
test.log
Directories:
UDAY/
backup_20251209/
bigfolder/
emptyhome/
missing-semester/
onlyfolders/
smallfolder/
testfolder/
Total size: 123180 KB
```

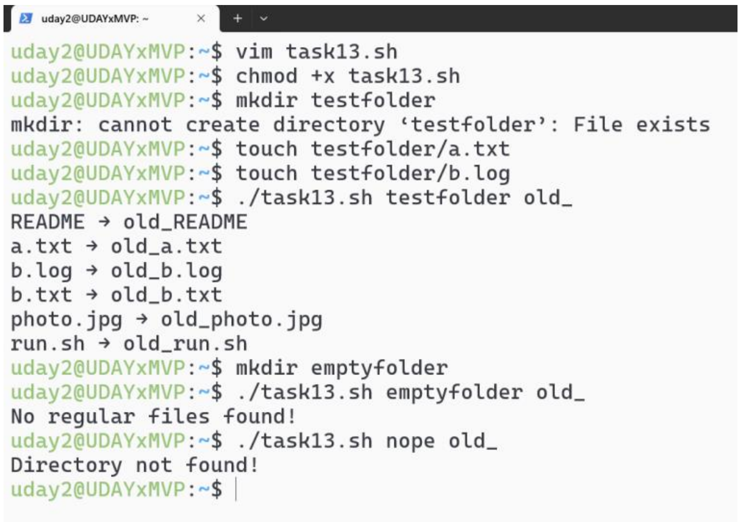
Task 13 – Bulk Rename with Prefix (Update)

Write a script that:

1. Takes **two arguments**:
   - A directory path.
   - A prefix string.
2. For every **regular file** directly inside that directory (non-recursive):
   - Rename the file by adding the prefix before its original name.
   - Example: if prefix is `old_` and file is `data.txt`, it becomes `old_data.txt`.
3. Print the old name and new name for each renamed file.
4. If the directory does not exist, or there are no regular files, print a suitable message.

ANSWER:

```
dir="$1"
pre="$2"

[ -d "$dir"] || {echo "Directory not found!"; exit 1;}
[ -z "$pre"] && {echo "Prefix missing!"; exit 1;}

count=0

for f in "$dir"/*; do
   [ -f "$f"] || continue
   name=$(basename "$f")
   mv "$f" "$dir/$pre$name"
   echo "$name → $pre$name"
   count=$((count+1))
done

[ "$count" -eq 0 ] && echo "No regular files found!"
```



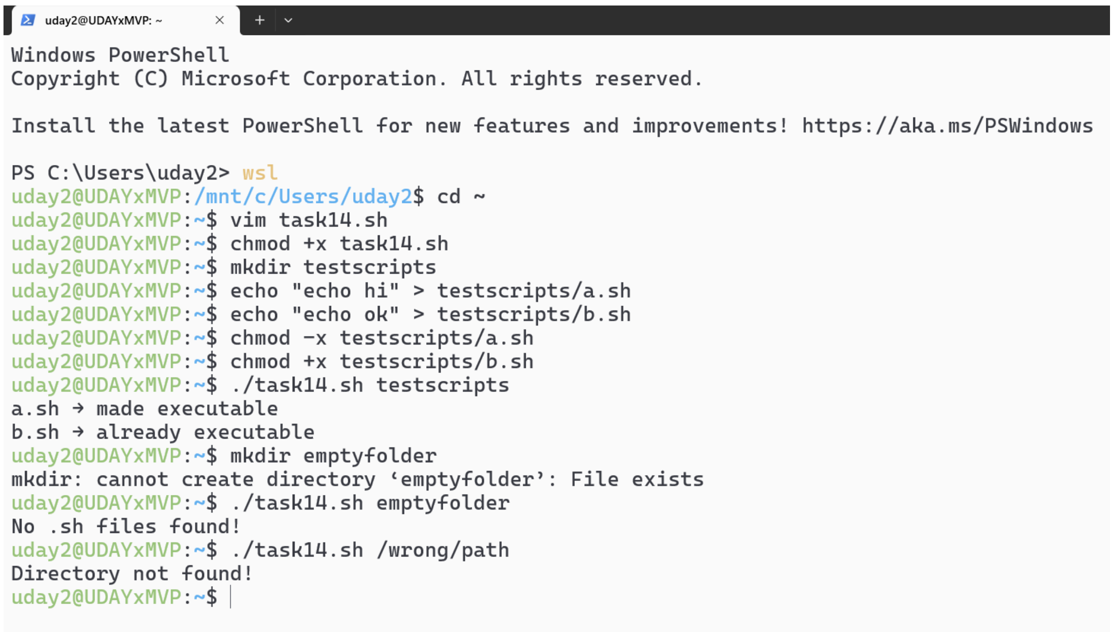
```
uday2@UDAYxMVP:~$ vim task13.sh
uday2@UDAYxMVP:~$ chmod +x task13.sh
uday2@UDAYxMVP:~$ mkdir testfolder
mkdir: cannot create directory ‘testfolder’: File exists
uday2@UDAYxMVP:~$ touch testfolder/a.txt
uday2@UDAYxMVP:~$ touch testfolder/b.log
uday2@UDAYxMVP:~$ ./task13.sh testfolder old_
README → old_README
a.txt → old_a.txt
b.log → old_b.log
b.txt → old_b.txt
photo.jpg → old_photo.jpg
run.sh → old_run.sh
uday2@UDAYxMVP:~$ mkdir emptyfolder
uday2@UDAYxMVP:~$ ./task13.sh emptyfolder old_
No regular files found!
uday2@UDAYxMVP:~$ ./task13.sh nope old_
Directory not found!
uday2@UDAYxMVP:~$
```

Task 14 – Permissions Updater for Scripts (Update)

Write a script that:

1. Takes **one argument**: a directory path.
2. For every **shell script file** (files ending with `.sh`) directly inside that directory:
   - If the file is **not executable**, make it executable for the user (e.g. add `u+x`).
3. Print the name of each `.sh` file and whether you changed its permissions or left it as is.
4. Handle the case where the directory does not exist or there are no `.sh` files.

```
dir="$1"
[ -d "$dir"] || {echo "Directory not found!"; exit 1;}

count=0

for f in "$dir"/*.sh; do
   [ -f "$f"] || continue
   count=$((count+1))
   if [! -x "$f"]; then
      chmod u+x "$f"
      echo "$(basename "$f") → made executable"
   else
      echo "$(basename "$f") → already executable"
   fi
done

[ "$count" -eq 0] && echo "No .sh files found!"
```



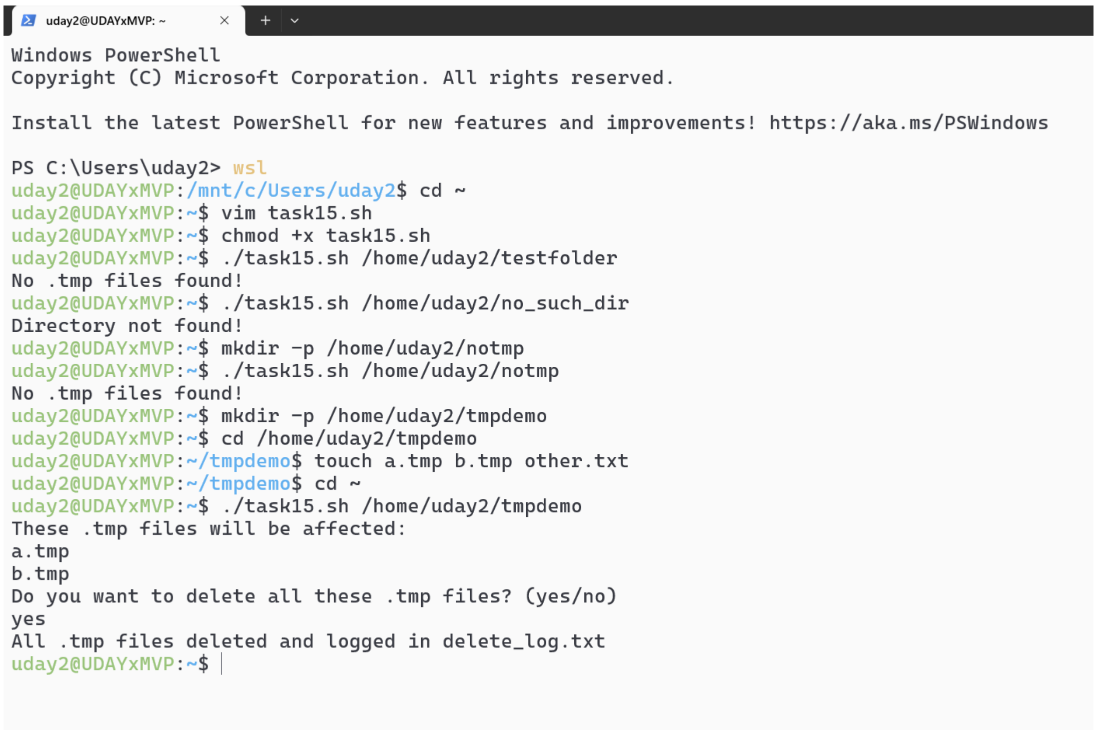
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\uday2> wsl
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$ vim task14.sh
uday2@UDAYxMVP:~$ chmod +x task14.sh
uday2@UDAYxMVP:~$ mkdir testscripts
uday2@UDAYxMVP:~$ echo "echo hi" > testscripts/a.sh
uday2@UDAYxMVP:~$ echo "echo ok" > testscripts/b.sh
uday2@UDAYxMVP:~$ chmod -x testscripts/a.sh
uday2@UDAYxMVP:~$ chmod +x testscripts/b.sh
uday2@UDAYxMVP:~$ ./task14.sh testscripts
a.sh → made executable
b.sh → already executable
uday2@UDAYxMVP:~$ mkdir emptyfolder
mkdir: cannot create directory ‘emptyfolder’: File exists
uday2@UDAYxMVP:~$ ./task14.sh emptyfolder
No .sh files found!
uday2@UDAYxMVP:~$ ./task14.sh /wrong/path
Directory not found!
uday2@UDAYxMVP:~$
```

Task 15 – Safe Cleaner with Delete Log (Delete)

Write a script that:

1. Takes **one argument**: a directory path.
2. Creates (if not already present) a file named `delete_log.txt` inside that directory.
3. Shows the user a list of all **`.tmp` files** in that directory (non-recursive).
4. Asks the user: "Do you want to delete all these .tmp files? (yes/no)".
    - If the user types `yes`:
        - Delete the `.tmp` files.
        - For each deleted file, append a line to `delete_log.txt` with the file name and the current date/time.
    - If the user types anything else, do not delete and just exit.
5. If there are no `.tmp` files, print a message and do nothing.

```
dir="$1"
if [ -z "$dir"]; then
  echo "Usage: $0 DIRECTORY"
  exit 1
fi

if [! -d "$dir"]; then
  echo "Directory not found!"
  exit 1
fi

cd "$dir" || exit 1

set -- *.tmp
if [ "$1" = "*.tmp"]; then
  echo "No .tmp files found!"
  exit 0
fi

echo "These .tmp files will be affected:"
for f in "$@"; do
  echo "$f"
done

echo "Do you want to delete all these .tmp files? (yes/no)"
read answer

if [ "$answer" != "yes"]; then
  echo "Aborted. No files deleted."
  exit 0
fi
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\uday2> wsl
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$ vim task15.sh
uday2@UDAYxMVP:~$ chmod +x task15.sh
uday2@UDAYxMVP:~$ ./task15.sh /home/uday2/testfolder
No .tmp files found!
uday2@UDAYxMVP:~$ ./task15.sh /home/uday2/no_such_dir
Directory not found!
uday2@UDAYxMVP:~$ mkdir -p /home/uday2/notmp
uday2@UDAYxMVP:~$ ./task15.sh /home/uday2/notmp
No .tmp files found!
uday2@UDAYxMVP:~$ mkdir -p /home/uday2/tmpdemo
uday2@UDAYxMVP:~$ cd /home/uday2/tmpdemo
uday2@UDAYxMVP:~/tmpdemo$ touch a.tmp b.tmp other.txt
uday2@UDAYxMVP:~/tmpdemo$ cd ~
uday2@UDAYxMVP:~$ ./task15.sh /home/uday2/tmpdemo
These .tmp files will be affected:
a.tmp
b.tmp
Do you want to delete all these .tmp files? (yes/no)
yes
All .tmp files deleted and logged in delete_log.txt
uday2@UDAYxMVP:~$
```

Task 16 – Recursive Extension Backup (Create/Read)

Write a script that:

1. Takes **two arguments**:
   - A directory path.
   - A file extension (for example: `txt`, `log`, `sh` – without the dot).
2. Creates a directory named `ext_backup` in the current directory if it doesn't exist.
3. Recursively searches the given directory for files with that extension.
4. Copies all matching files into `ext_backup` while preserving only the file name (not the directory structure).
5. Prints how many files were copied.

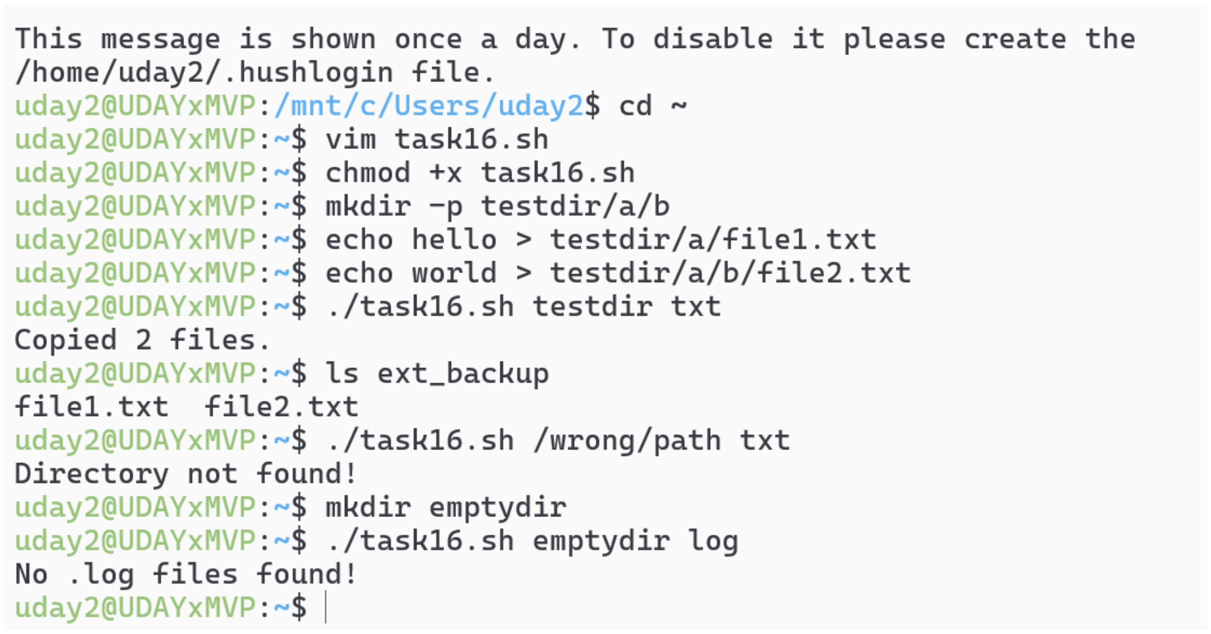
Handle the cases where:

- The directory does not exist.

No files with the given extension are found.

ANSWER:

```
dir="$1"
ext="$2"

[ -d "$dir" ] || { echo "Directory not found!"; exit 1; }
[ -z "$ext" ] && { echo "Extension missing!"; exit 1; }

mkdir -p ext_backup

count=0

for f in $(find "$dir" -type f -name "*.$ext"); do
    cp "$f" "ext_backup/$(basename "$f")"
    count=$((count+1))
done

[ "$count" -eq 0 ] && { echo "No .$ext files found!"; exit 0; }
```



```
This message is shown once a day. To disable it please create the
/home/uday2/.hushlogin file.
uday2@UDAYxMVP:/mnt/c/Users/uday2$ cd ~
uday2@UDAYxMVP:~$ vim task16.sh
uday2@UDAYxMVP:~$ chmod +x task16.sh
uday2@UDAYxMVP:~$ mkdir -p testdir/a/b
uday2@UDAYxMVP:~$ echo hello > testdir/a/file1.txt
uday2@UDAYxMVP:~$ echo world > testdir/a/b/file2.txt
uday2@UDAYxMVP:~$ ./task16.sh testdir txt
Copied 2 files.
uday2@UDAYxMVP:~$ ls ext_backup
file1.txt  file2.txt
uday2@UDAYxMVP:~$ ./task16.sh /wrong/path txt
Directory not found!
uday2@UDAYxMVP:~$ mkdir emptydir
uday2@UDAYxMVP:~$ ./task16.sh emptydir log
No .log files found!
uday2@UDAYxMVP:~$ 
```

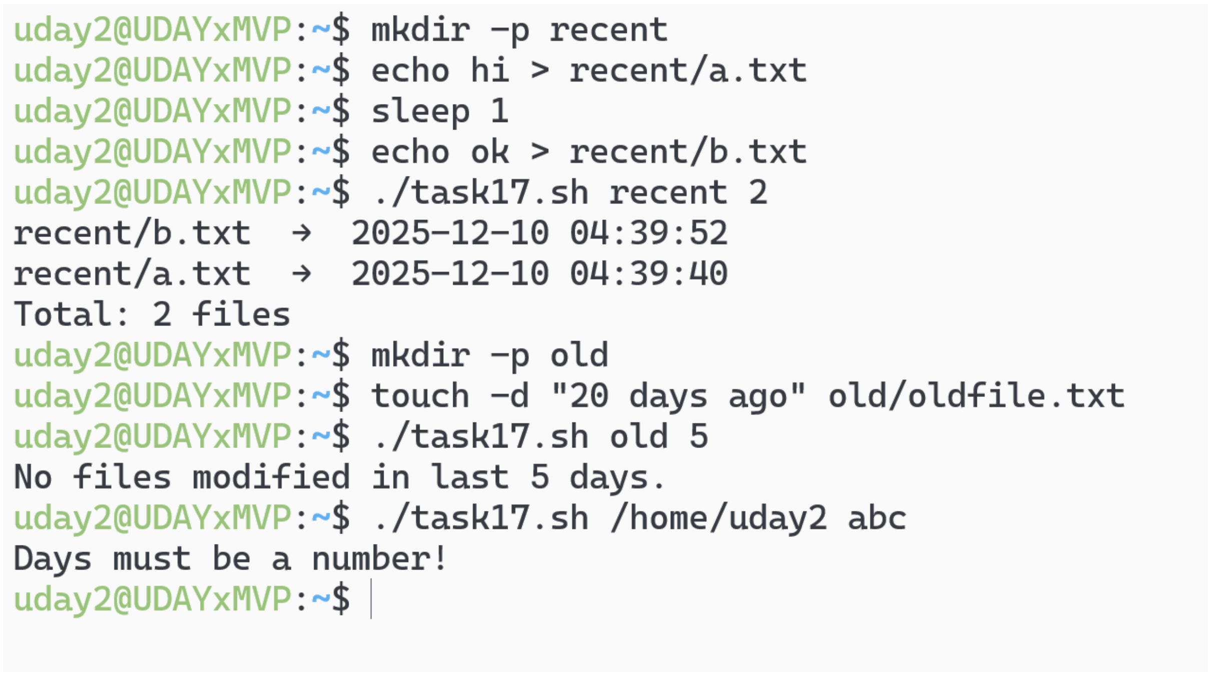
Task 17 – Recent Files Reporter (Read)

Write a script that:

1. Takes **two arguments**:
    - A directory path.
    - A number of days `N`.
2. Recursively finds all files under that directory that were **modified in the last N days**.
3. Prints the path of each file and its last modified time.
4. At the end, prints the total count of such files.

If `N` is missing or not a number, print an error. If no such files are found, show a message.

```
#!/bin/bash

dir="$1"
days="$2"

[ -d "$dir" ] || { echo "Directory not found!"; exit 1; }
[[ "$days" =~ ^[0-9]+$ ]] || { echo "Days must be a number!"; exit 1; }

count=0

while IFS= read -r f; do
    ts=$(date -r "$f" "+%Y-%m-%d %H:%M:%S")
    echo "$f  →  $ts"
    count=$((count+1))
done < <(find "$dir" -type f -mtime "-$days")
```



```
uday2@UDAYxMVP:~$ mkdir -p recent
uday2@UDAYxMVP:~$ echo hi > recent/a.txt
uday2@UDAYxMVP:~$ sleep 1
uday2@UDAYxMVP:~$ echo ok > recent/b.txt
uday2@UDAYxMVP:~$ ./task17.sh recent 2
recent/b.txt  →  2025-12-10 04:39:52
recent/a.txt  →  2025-12-10 04:39:40
Total: 2 files
uday2@UDAYxMVP:~$ mkdir -p old
uday2@UDAYxMVP:~$ touch -d "20 days ago" old/oldfile.txt
uday2@UDAYxMVP:~$ ./task17.sh old 5
No files modified in last 5 days.
uday2@UDAYxMVP:~$ ./task17.sh /home/uday2 abc
Days must be a number!
uday2@UDAYxMVP:~$
```

Task 18 – Merge Text Files (Create/Read)

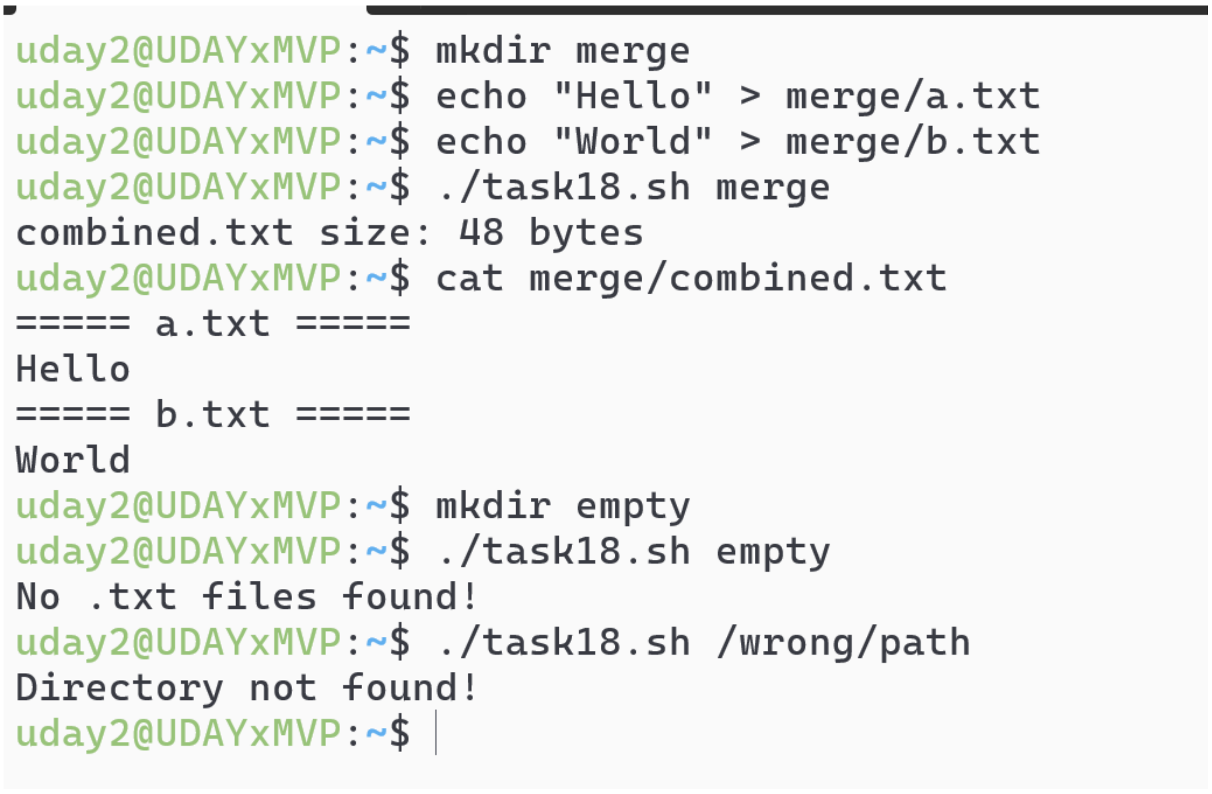
Write a script that:

1. Takes **one argument**: a directory path.
2. Looks for all `.txt` files directly inside that directory (non-recursive).
3. Creates a new file named `combined.txt` inside that directory.
4. For each `.txt` file, in alphabetical order:
    - Writes a header line into `combined.txt` like:
        - `===== FILENAME.txt =====`
    - Then appends the content of that file.

5. At the end, prints the size (in bytes) of `combined.txt`.

If there are no `.txt` files, print a message and do not create `combined.txt`.

```
#!/bin/bash

dir="$1"
[ -d "$dir" ] || { echo "Directory not found!"; exit 1; }

txts=($(ls "$dir"/*.txt 2>/dev/null))
[ ${#txts[@]} -eq 0 ] && { echo "No .txt files found!"; exit 0; }

out="$dir/combined.txt"
> "$out"

for f in "${txts[@]}"; do
   name=$(basename "$f")
   echo "===== $name =====" >> "$out"
   cat "$f" >> "$out"
done
```



```
uday2@UDAYxMVP:~$ mkdir merge
uday2@UDAYxMVP:~$ echo "Hello" > merge/a.txt
uday2@UDAYxMVP:~$ echo "World" > merge/b.txt
uday2@UDAYxMVP:~$ ./task18.sh merge
combined.txt size: 48 bytes
uday2@UDAYxMVP:~$ cat merge/combined.txt
===== a.txt =====
Hello
===== b.txt =====
World
uday2@UDAYxMVP:~$ mkdir empty
uday2@UDAYxMVP:~$ ./task18.sh empty
No .txt files found!
uday2@UDAYxMVP:~$ ./task18.sh /wrong/path
Directory not found!
uday2@UDAYxMVP:~$
```

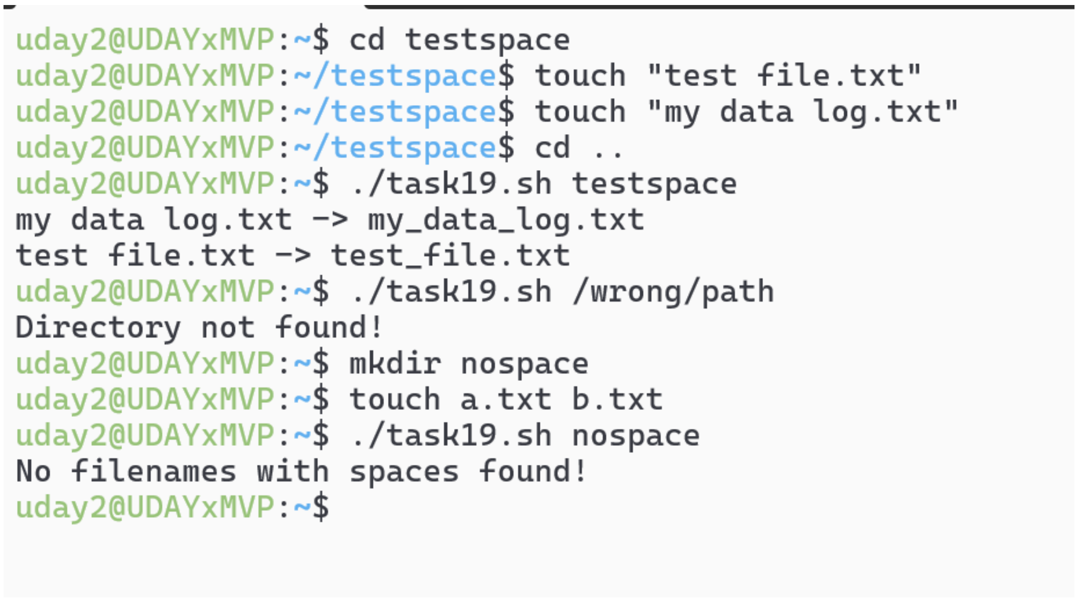
Task 19 – Whitespace Renamer (Update)

Write a script that:

1. Takes **one argument**: a directory path.
2. For every **regular file** directly inside that directory whose name contains spaces:
    - Rename the file by replacing all spaces with underscores (_).
    - Example: `my file name.txt` → `my_file_name.txt`.
3. Print each change as: `old_name -> new_name`.
4. If no filenames contain spaces, print a message.

Be careful not to accidentally modify directories in this task.

```
#!/bin/bash

dir="$1"
[ -d "$dir" ] || { echo "Directory not found!"; exit 1; }

count=0

for f in "$dir"/*; do
    [ -f "$f" ] || continue
    name=$(basename "$f")
    new="${name// /_}"
    if [ "$name" != "$new" ]; then
        mv "$dir/$name" "$dir/$new"
        echo "$name -> $new"
        count=$((count+1))
    fi
done

[ "$count" -eq 0 ] && echo "No filenames with spaces found!"
```


```
uday2@UDAYxMVP:~$ cd testspace
uday2@UDAYxMVP:~/testspace$ touch "test file.txt"
uday2@UDAYxMVP:~/testspace$ touch "my data log.txt"
uday2@UDAYxMVP:~/testspace$ cd ..
uday2@UDAYxMVP:~$ ./task19.sh testspace
my data log.txt -> my_data_log.txt
test file.txt -> test_file.txt
uday2@UDAYxMVP:~$ ./task19.sh /wrong/path
Directory not found!
uday2@UDAYxMVP:~$ mkdir nospace
uday2@UDAYxMVP:~$ touch a.txt b.txt
uday2@UDAYxMVP:~$ ./task19.sh nospace
No filenames with spaces found!
uday2@UDAYxMVP:~$
```

Task 20 – Note Manager Using a Directory (CRUD)

Write a script that manages notes using a directory named `notes` in the current folder.

The script should:

1. Create the `notes` directory if it does not exist.
2. Show this menu:
   1. Create a new note
   2. List all notes
   3. View a note
   4. Delete a note
   5. Exit
3. **Create a new note**:
   - Ask for a note title (use it as the file name, e.g. `title.txt`).
   - Ask the user to type a line of text and save it into that file inside `notes/`.

List all notes:

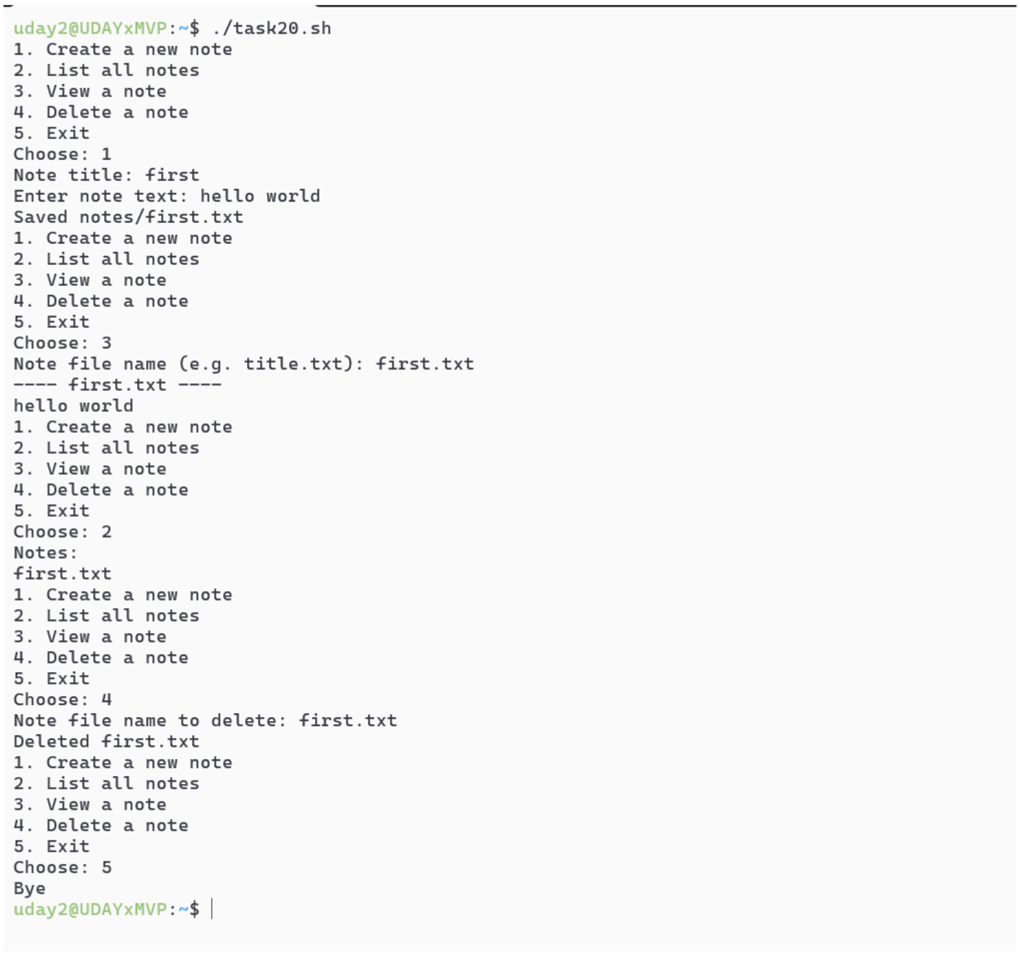
- Show all files inside `notes/`.

View a note:

- Ask for the note file name and display its content if it exists.

Delete a note:

- Ask for the note file name and delete it if it exists.



```
uday2@UDAYxMVP:~$ ./task20.sh
1. Create a new note
2. List all notes
3. View a note
4. Delete a note
5. Exit
Choose: 1
Note title: first
Enter note text: hello world
Saved notes/first.txt
1. Create a new note
2. List all notes
3. View a note
4. Delete a note
5. Exit
Choose: 3
Note file name (e.g. title.txt): first.txt
---- first.txt ----
hello world
1. Create a new note
2. List all notes
3. View a note
4. Delete a note
5. Exit
Choose: 2
Notes:
first.txt
1. Create a new note
2. List all notes
3. View a note
4. Delete a note
5. Exit
Choose: 4
Note file name to delete: first.txt
Deleted first.txt
1. Create a new note
2. List all notes
3. View a note
4. Delete a note
5. Exit
Choose: 5
Bye
uday2@UDAYxMVP:~$
```

```
mkdir -p notes

while true; do
  echo "1. Create a new note"
  echo "2. List all notes"
  echo "3. View a note"
  echo "4. Delete a note"
  echo "5. Exit"
  read -p "Choose: " choice

  case "$choice" in
    1)
      read -p "Note title: " title
      file="notes/$title.txt"
      read -p "Enter note text: " line
      echo "$line" > "$file"
      echo "Saved $file"
      ;;
    2)
      if [ "$(ls -A notes)" ]; then
        echo "Notes:"
        ls notes
      else
        echo "No notes yet"
      fi
      ;;
    3)
      read -p "Note file name (e.g. title.txt): " name
      file="notes/$name"
      if [ -f "$file" ]; then
        echo "---- $name ----"
        cat "$file"
      else
        echo "Note not found"
      fi
      ;;
    4)
      read -p "Note file name to delete: " name
      file="notes/$name"
      if [ -f "$file" ]; then
        rm "$file"
        echo "Deleted $name"
      else
        echo "Note not found"
      fi
      ;;
    5)
      echo "Bye"
      break
      ;;
    *)
      echo "Invalid choice"
      ;;
  esac
done
```