**`Easy Problems with Solutions:**

1. Problem: Write a shell script that takes two numbers as input and outputs their sum.

Solution:

#!/bin/bash

echo "Enter the first number: "

read num1

echo "Enter the second number: "

read num2

sum=$((num1 + num2))

echo "The sum is: $sum"

1. Problem: Write a shell script that lists all the files in a directory.

Solution:

#!/bin/bash

echo "Files in the current directory:"

ls

1. Problem: Write a shell script that checks whether a given number is even or odd.

Solution:

#!/bin/bash

echo "Enter a number: "

read num

if ((num % 2 == 0)); then

echo "$num is even"

else

echo "$num is odd"

fi

1. Problem: Write a shell script that displays the current date and time.

Solution:

#!/bin/bash

echo "Current date and time:"

date

1. Problem: Write a shell script that counts the number of lines in a given file.

Solution:

#!/bin/bash

echo "Enter the file name: "

read filename

line\_count=$(wc -l < "$filename")

echo "The file $filename has $line\_count lines"

**Exercises for Shell Scripting:**

1. Write a shell script that takes a file name as input and checks whether the file exists or not. If it exists, display a message saying "File exists", otherwise display "File does not exist."
2. Write a shell script that prompts the user to enter their name, and then greets them with a personalized message.
3. Write a shell script that calculates the factorial of a given number. The script should take the number as input and display the factorial.
4. Write a shell script that renames all files in a directory by adding a prefix "new\_" to their names. For example, if there are files named "file1.txt" and "file2.txt", the script should rename them to "new\_file1.txt" and "new\_file2.txt".
5. Write a shell script that takes a sentence as input and displays the number of words in the sentence.

Feel free to try out these problems and exercises!

**Intermediate/Medium Problems with Solutions:**

1. Problem: Write a shell script that takes a directory name as input and counts the number of files and subdirectories in that directory.

Solution:

#!/bin/bash

echo "Enter the directory name: "

read dirname

file\_count=$(find "$dirname" -type f | wc -l)

dir\_count=$(find "$dirname" -type d | wc -l)

echo "Number of files: $file\_count"

echo "Number of subdirectories: $dir\_count"

1. Problem: Write a shell script that prompts the user to enter a number and calculates the sum of all numbers from 1 to that number.

Solution:

#!/bin/bash

echo "Enter a number: "

read num

sum=0

for ((i = 1; i <= num; i++)); do

sum=$((sum + i))

done

echo "The sum is: $sum"

1. Problem: Write a shell script that checks whether a given string is a palindrome or not.

Solution:

#!/bin/bash

echo "Enter a string: "

read str

reverse=$(echo "$str" | rev)

if [ "$str" = "$reverse" ]; then

echo "$str is a palindrome"

else

echo "$str is not a palindrome"

fi

1. Problem: Write a shell script that finds and displays all the files in a directory that have a specific extension (e.g., ".txt").

Solution:

#!/bin/bash

echo "Enter the directory name: "

read dirname

echo "Enter the file extension: "

read extension

find "$dirname" -type f -name "\*$extension"

1. Problem: Write a shell script that displays the size of the largest file in a directory.

Solution:

#!/bin/bash

echo "Enter the directory name: "

read dirname

largest\_file=$(find "$dirname" -type f -exec du -ch {} + | grep -e "^[0-9.]\*[MG]" | sort -rh | head -1)

echo "The largest file in $dirname is: $largest\_file"

**Exercises for Shell Scripting:**

1. Write a shell script that prompts the user to enter a directory name and recursively lists all files and subdirectories in that directory.
2. Write a shell script that takes a sentence as input and displays the number of vowels and consonants in the sentence.
3. Write a shell script that prompts the user to enter a directory name and recursively delete all empty subdirectories in that directory.
4. Write a shell script that takes a filename as input and counts the number of words, lines, and characters in that file.
5. Write a shell script that displays a menu with options for creating, renaming, or deleting files in a directory. Implement the corresponding actions for each option.

**Advanced Problems with Solutions:**

1. Problem: Write a shell script that recursively finds and counts the number of files in a directory, including all its subdirectories, and displays the count for each subdirectory.

Solution:

#!/bin/bash

count\_files() {

local directory=$1

local count=0

for file in "$directory"/\*; do

if [[ -f "$file" ]]; then

count=$((count + 1))

elif [[ -d "$file" ]]; then

subdirectory\_count=$(count\_files "$file")

count=$((count + subdirectory\_count))

fi

done

echo "$directory: $count"

return $count

}

echo "Enter the directory name: "

read dirname

count\_files "$dirname"

1. Problem: Write a shell script that finds and displays the top 5 largest files in a directory and its subdirectories.

Solution:

#!/bin/bash

echo "Enter the directory name: "

read dirname

find "$dirname" -type f -exec du -ah {} + | sort -rh | head -n 5

1. Problem: Write a shell script that takes a directory name as input and recursively finds and lists all files that have been modified within the last 24 hours.

Solution:

#!/bin/bash

echo "Enter the directory name: "

read dirname

find "$dirname" -type f -mtime -1

1. Problem: Write a shell script that takes a file name as input and replaces all occurrences of a given word with another word.

Solution:

#!/bin/bash

echo "Enter the file name: "

read filename

echo "Enter the word to replace: "

read word1

echo "Enter the new word: "

read word2

sed -i "s/$word1/$word2/g" "$filename"

echo "Replacement complete."

1. Problem: Write a shell script that monitors a log file in real-time and displays new lines as they are added to the file.

Solution:

#!/bin/bash

echo "Enter the log file name: "

read logfile

tail -f "$logfile"

**Exercises for Shell Scripting**:

1. Write a shell script that takes a directory name as input and creates a compressed archive (e.g., tar.gz) of all the files in the directory, excluding subdirectories.
2. Write a shell script that prompts the user to enter a directory name and finds all duplicate files (based on content) within that directory.
3. Write a shell script that takes a file name as input and encrypts its contents using a simple substitution cipher. The script should prompt the user to enter the substitution key.
4. Write a shell script that monitors the CPU usage of a specific process and sends an email notification if the usage exceeds a certain threshold.
5. Write a shell script that takes a directory name as input and recursively finds and displays the total disk space used by each user within that directory.