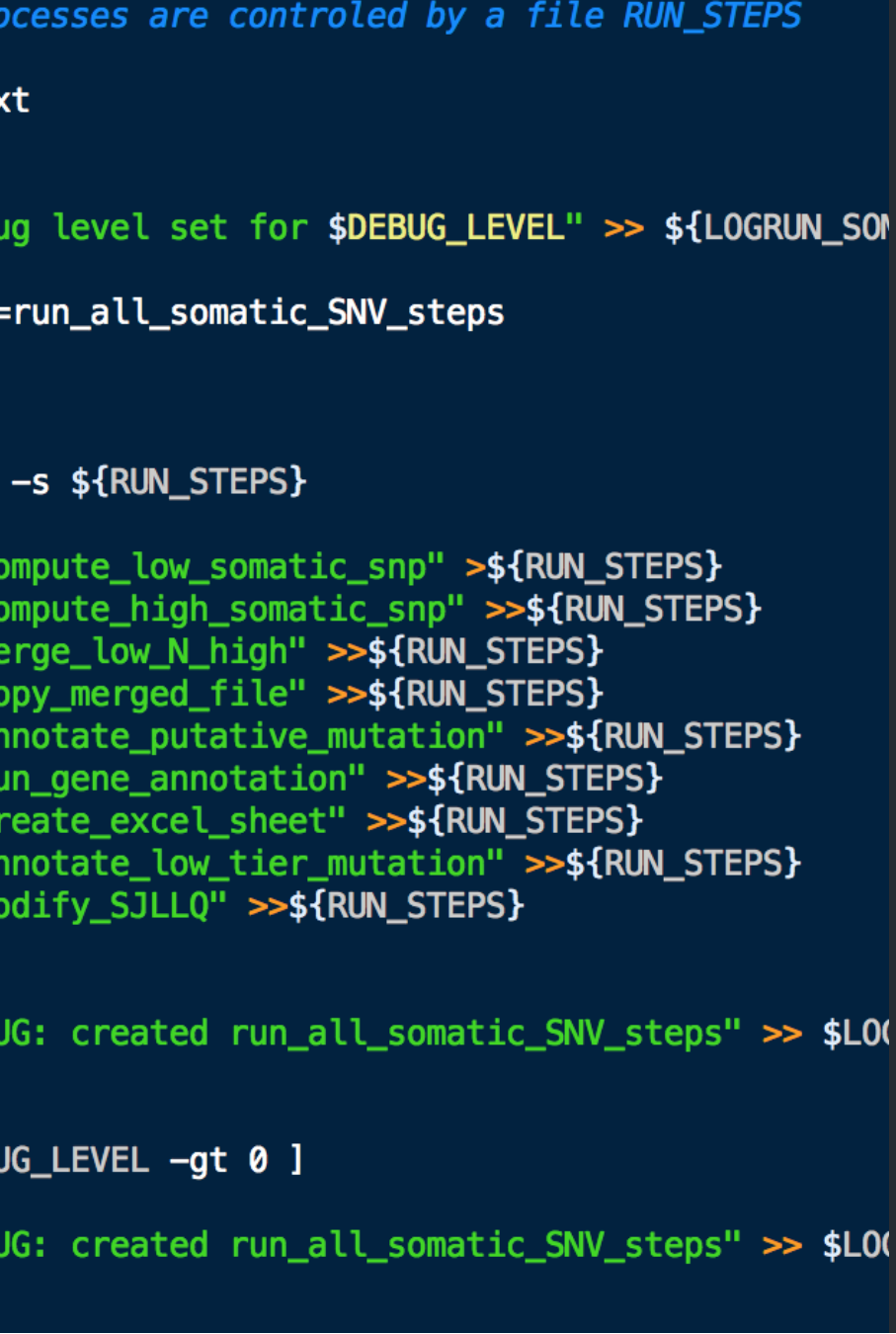



# Shell Scripting Basics

Learn the fundamentals of shell scripting with these practical examples. Explore different techniques to manipulate files and directories using Bash commands.

# Extracting Lines from a File

## Question

Q1) Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.

## Code

```
echo "Enter file name:" read f echo "Enter the starting line :" read s echo "Enter the ending line :" read e sed -n $s,$e\p $f
```

## Output

```
Enter file name: a.txt Enter the starting line : 2 Enter the ending line : 5 Result: i am new to
```

# Deleting Lines from Files

## Question

Q2) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

## Code

```
echo "filename:" read f echo "word:" read w echo "file before removing $w" cat $f grep -v $w $f > t mv t $f echo "file after removing $w" cat $f
```

## Output

```
filename: a.txt word: hi file having content before removing word hi is hr i am new to this linux programing file having content after removing word hi is i am new to linux programing
```

# Listing Files in a Directory

## Question

Q3) Write a shell script that displays a list of all the files in the current directory.

## Code

```
echo $(ls)
```

## Output

```
a.txt b.txt s2.sh s3.sh s4.sh s.sh
```

# Determining File Types

## Question

Q4) Write a shell script that receives any number of file names as arguments check if every argument supplied is a file or a directory and reports accordingly.

## Code

```
read t while [ $t -ge 1 ]; do read x if [ -f $x ] then echo "it is File" elif [ -d $x ] then echo " it is directory" else echo "it is neither of them" fi ((t=t-1)) done
```

## Output

```
4 a.txt it is File b.txt it is File Desktop it is neither of them jk it is neither of them
```

# Searching and Replacing Text

1

## Question

Q5) Write a shell script that searches for a specified word in a file and replaces it with another word.

2

## Code

```
echo "filename:" read f echo "search word:" read sw echo "replace word:" read rw sed -i "s/$sw/$rw/g" $f
```

3

## Output

filename: test.txt search word: hello replace word: hi Contents of test.txt before replacement: hello world! hello beautiful! Contents of test.txt after replacement: hi world! hi beautiful!

# Creating Directories and Files

## Question

Q6) Write a shell script that asks the user for a directory name and creates it if it doesn't exist. If the directory already exists, ask the user if they want to create a file inside it.

## Code

```
echo "Enter directory name:" read dirname if [ ! -d $dirname ]; then mkdir $dirname echo "Directory created successfully." else echo "Directory already exists. Do you want to create a file inside it? (y/n)" read response if [ $response = "y" ]; then echo "Enter file name:" read filename touch $dirname/$filename echo "File created successfully." else echo "No file created." fi fi
```

## Output

Enter directory name: docs Directory created successfully. Enter directory name: docs Directory already exists. Do you want to create a file inside it? (y/n) y Enter file name: notes.txt File created successfully.

# Counting Words in Files

## Question

Q7) Write a shell script that counts the number of words in a given file.

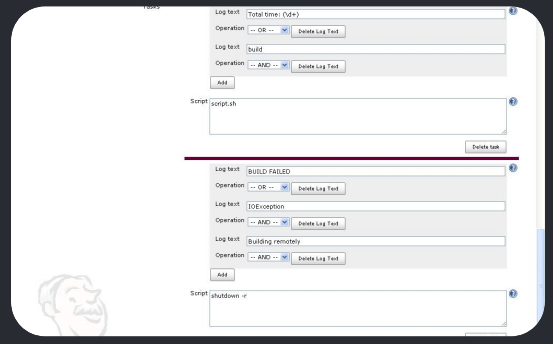
## Code

```
echo "Enter file name:" read filename count=$(wc -w $filename | awk '{print $1}') echo "Number of words in $filename: $count"
```

## Output

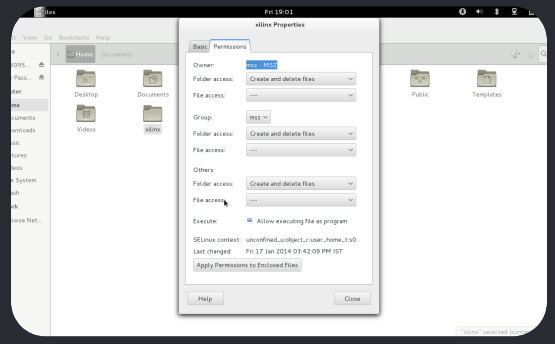
```
Enter file name: text.txt
Number of words in text.txt: 125
```

# Executing Shell Scripts



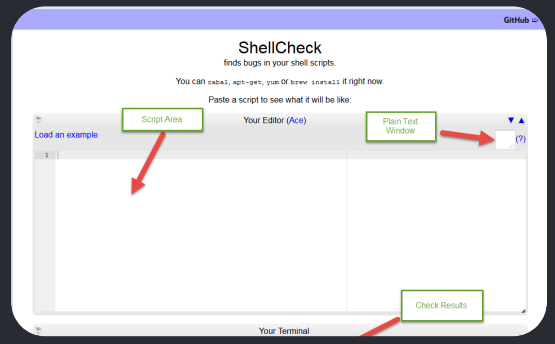
## Running Scripts

To execute a shell script, open a terminal and navigate to the directory containing the script. Use the following command: `./script.sh`



## Setting Permissions

Before executing a shell script, make sure it has the necessary permissions. Use the `chmod` command to allow execution: `chmod +x script.sh`




## Debugging Scripts

If you encounter errors while running a script, you can use the `set -x` command to enable debugging mode. This will print each command as it is executed.