# Grep with Pipe

Here are some **interview questions** involving scenarios where `grep` is combined with pipes, testing your ability to use `grep` effectively in a pipeline of commands:

---

### 1. Scenario: Search for a Pattern and Count the Number of Matches

**Question:** How can you search for the word "error" in a log file and count the number of lines that contain this word?

**Solution:**
Use `grep` to search for the pattern and pipe it to `wc -l` to count the lines.

```
grep "error" logfile.log | wc -l
```

---

### 2. Scenario: Search and Sort Unique Matches

**Question:** You want to search for all occurrences of the word "user" in a file and display only the unique matching lines, sorted alphabetically. How would you do this?

**Solution:**
Pipe the results of `grep` to `sort` and `uniq`.

```
grep "user" filename.txt | sort | uniq
```

---

### 3. Scenario: Search Through a Command's Output

**Question:** How can you display all running processes that contain the word "java" using `ps` and `grep`?

**Solution:**
Use `ps` to list processes and pipe the output to `grep`.

```
ps aux | grep "java"
```

---

### 4. Scenario: Search and Limit the Output

**Question:** You want to search for the word "failed" in a large file but only display the first 5 matching lines. What command would you use?

**Solution:**
Pipe the results of `grep` to `head`.

```
grep "failed" logfile.log | head -n 5
```

---

### 5. Scenario: Search for a Pattern and Exclude Certain Results

**Question:** You are searching for the word "http" in a configuration file but want to exclude lines that contain "https". How would you do this?

**Solution:**

Use two `grep` commands: one to include and one to exclude patterns.

```
grep "http" config.txt | grep -v "https"
```

---

### 6. Scenario: Extract Specific Columns from a Search Result

**Question:** You need to search for the word "session" in a file and extract only the second column from the matching lines. How would you achieve this?

**Solution:**

Pipe the output of `grep` to `awk` to extract specific columns.

```
grep "session" file.txt | awk '{print $2}'
```

---

### 7. Scenario: Search for a Word and Display the Number of Matches in Each File

**Question:** You need to search for the word "timeout" in multiple files and display the count of matches in each file. How would you do this?

**Solution:**

Use `grep -c` and combine it with `xargs` to process multiple files.

```
grep -c "timeout" *.txt | xargs -I {} echo {}
```

---

### 8. Scenario: Search and Display Matching Files Only

**Question:** How would you search for the word "error" across all files in a directory and only display the filenames that contain a match?

**Solution:**

Pipe the output of `grep` to `cut` to extract filenames.

```
grep -l "error" /path/to/files/*.log | cut -d ":" -f1
```

---

### 9. Scenario: Filter Command Output for Specific Keywords

**Question:** You need to list all files in a directory and filter out the ones that contain the word "temp". What command would you use?

**Solution:**

Use `ls` to list files and pipe it to `grep -v` to exclude matches.

```
ls | grep -v "temp"
```

---

### 10. Scenario: Search for Files and Filter by Word

**Question:** You want to find all `.sh` files in a directory and filter those that contain the word "backup". How do you do this using `find` and `grep`?

**Solution:**

Use `find` to locate `.sh` files and pipe the result to `grep` to filter content.

```
find /path/to/search -name "*.sh" | xargs grep "backup"
```

---

### 11. Scenario: Search Through Environment Variables

**Question:** How would you search for environment variables that contain the string "PATH"?

**Solution:**
Use `env` to list environment variables and pipe it to `grep`.

```
env | grep "PATH"
```

---

### 12. Scenario: Search for Words in a Compressed File

**Question:** How do you search for the word "error" in a compressed `.gz` log file and display only matching lines?

**Solution:**
Pipe the output of `zcat` to `grep`.

```
zcat logfile.gz | grep "error"
```

---

### 13. Scenario: Search for a Pattern and Sort by Frequency

**Question:** How can you search for all email addresses in a file and sort them by frequency of occurrence?

**Solution:**
Use `grep` to find the email addresses, then pipe it to `sort` and `uniq -c`.

```
grep -oE '\\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Z|a-z]{2,7}\\b' file.txt | sort | uniq -c | sort -nr
```

---

### 14. Scenario: Display a List of Unique Matching Strings

**Question:** You want to extract all IP addresses from a file and display only the unique ones. How would you achieve this?

**Solution:**
Use `grep` to find the IP addresses and pipe it to `uniq`.

```
grep -oE '([0-9]{1,3}\\.){3}[0-9]{1,3}' file.txt | sort | uniq
```

---

### 15. Scenario: Chain Multiple Grep Commands

**Question:** How can you search for lines that contain both "failed" and "user" in a file?

**Solution:**
Pipe the result of one `grep` command into another.

```
grep "failed" logfile.log | grep "user"
```

---

### 16. Scenario: Search Through Log Files and Filter Date Range

**Question:** How would you search for error logs from the output of `grep` but exclude entries that occurred in 2023?

**Solution:**

Pipe the output of `grep` to another `grep -v` to exclude the specific date range.

```
grep "error" logfile.log | grep -v "2023"
```

---

### 17. Scenario: Display Non-Matching Lines

**Question:** How do you list all users from the `/etc/passwd` file except the ones with "nologin" as their shell?

**Solution:**

Use `grep -v` to exclude the "nologin" entries.

```
cat /etc/passwd | grep -v "/nologin"
```

---

### 18. Scenario: Filter Disk Usage Report

**Question:** You want to display the disk usage for all directories except those containing "tmp". How would you achieve this?

**Solution:**

Pipe the output of `du` to `grep -v` to exclude unwanted directories.

```
du -h /path/to/dirs | grep -v "tmp"
```

---

### 19. Scenario: Filter Command Output Based on Multiple Criteria

**Question:** How can you list all files in a directory that contain either "config" or "backup" in their names?

**Solution:**

Use `ls` and pipe it to `grep` with extended regular expressions.

```
ls | grep -E "config|backup"
```

---

### 20. Scenario: Search for Lines and Format Output

**Question:** You need to search for the word "password" in a configuration file and display the matching lines in uppercase. How would you do this?

**Solution:**

Pipe the output of `grep` to `tr` to convert the text to uppercase.

```
grep "password" config.txt | tr '[:lower:]' '[:upper:]'
```

---

These questions demonstrate how `grep` can be combined with pipes to filter, manipulate, and refine the output from other commands or utilities, showcasing its versatility in real-world scenarios.