Notes 7

Explain how to use each of the following commands.

cat

```
- Concatenates and displays the contents of files.
```

Formula:

```
- cat [options] [file...]
```

** Examples:

```
- cat file.txt → Display contents of file.txt.
- cat file1.txt file2.txt → Show contents of both files one after the
  other.
- cat > newfile.txt → Create a new file; input ends with CTRL+D.
```

tac

```
- Displays the contents of a file in reverse (last line first).
```

Formula:

```
- tac [file]
```

Examples:

```
- tac file.txt → Show file with lines in reverse order.
- tac file1.txt > reversed.txt → Save reversed content to a new file.
- tac log.txt | grep "ERROR" → Find errors in reverse log order.
```

head

```
 - Displays the first n lines of a file (default is 10).
```

Formula:

```
 - head [options] [file]
```

Examples:

```
 - head file.txt → First 10 lines of file.txt.
 - head -n 5 file.txt → First 5 lines.
 - head -c 100 file.txt → First 100 bytes.
```

tail

```
 - Displays the last n lines of a file (default is 10).
```

Formula:

```
 - tail [options] [file]
```

Examples:

```
 - tail file.txt → Last 10 lines.
 - tail -n 20 file.txt → Last 20 lines.
 - tail -f log.txt → Continuously monitor a file (useful for logs).
```

cut

```
 - Cuts and extracts sections from lines of text.
```

Formula:

```
- cut [options] [file]
```

Examples:

```
- cut -d',' -f1 names.csv → First column using comma delimiter.
- cut -c1-5 file.txt → First 5 characters of each line.
- cut -f2 -d':' /etc/passwd → Get second field from colon-delimited file.
```

---

sort

```
- Sorts lines in a file or input.
```

Formula:

```
- sort [options] [file]
```

Examples:

```
- sort names.txt → Alphabetically sort the file.
- sort -r numbers.txt → Reverse sort.
- sort -n scores.txt → Numeric sort.
```

---

wc

```
- Counts words, lines, characters, or bytes.
```

Formula:

```
- wc [options] [file]
```

Examples:

```
- wc file.txt → Lines, words, and bytes in file.
- wc -l file.txt → Count lines only.
- wc -w file.txt → Count words only.
```

tr

```
-  Translates or deletes characters.
```

Formula:

```
- tr [options] SET1 [SET2]
```

Examples:

```
- echo "hello" | tr a-z A-Z → Convert to uppercase.
- tr -d '0-9' < file.txt → Remove digits from file.
- tr ' ' '\n' < words.txt → Replace spaces with newlines.
```

diff

```
- Compares two files line by line.
```

Formula:

```
-  diff [options] file1 file2
```

Examples:

```
- diff file1.txt file2.txt → Show differences.
- diff -u old.txt new.txt → Unified format (used in patches).
- diff -y file1.txt file2.txt → Side-by-side comparison.
```

grep

- Searches for lines matching a pattern.

Formula:

- grep [options] PATTERN [file...]

Examples:

- grep "error" log.txt → Find lines containing "error".
- grep -i "fail" file.txt → Case-insensitive search.
- grep -r "TODO" ./src → Recursively search in directory.