Notes 5: Handling files

1. Cat

Definition

The `cat` command is used to concatenate and display the content of files.

Usage/Formula

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

- **Examples**

  1. Display the content of a single file:

  ```
  cat file.txt
  ```

  2. Concatenate multiple files and display their content:

  ```
  cat file1.txt file2.txt
  ```

  3. Display content with line numbers:

  ```
  cat -n file.txt
  ```

2. Tac

Definition:

The `tac` command is similar to `cat`, but it displays the content of files in reverse order.

Usage/Formula:

- `tac [options] [file ...]`

- **Examples**:

  1. Display content of a file in reverse order:

  ```
  tac file.txt
  ```

  2. Display content of multiple files in reverse order:

  ```
  tac file1.txt file2.txt
  ```

3. Head

- ### Definition:

- The `head` command is used to display the beginning of a file.

Usage/Formula:

- `head [options] [file ...]`
- **Examples**:
  1. Display the first 10 lines of a file:

     ```
     head file.txt
     ```

  2. Display the first 5 lines of a file:

```
head -n 5 file.txt
```

4. Tail

Definition:

The `tail` command is used to display the end of a file.

Usage/Formula:

```
tail [options] [file ...]
```

- **Examples**:
  1. Display the last 10 lines of a file:

```
tail file.txt
```

  2. Display the last 5 lines of a file:

```
tail -n 5 file.txt
```

5. Cut

Definition:

The `cut` command is used to extract sections from each line of files.

Usage/Formula:

```
cut [options] [file ...]
```

- **Examples**:
  1. Extract the first field (column) from a CSV file:

```
cut -d ',' -f 1 file.csv
```

  2. Extract a range of characters from each line:

```
cut -c 1-5 file.txt
```

6. Sort

Definition :

The `sort` command is used to sort lines of text files.

Usage/Formula:

```
sort [options] [file ...]
```

- **Examples**:
    1. Sort lines of a file alphabetically:

```
sort file.txt
```

2. Sort lines numerically (ascending):

```
sort -n file.txt
```

7. WC

Definition:

The `wc` command is used to count lines, words, and characters in a file.

Usage/Formula:

```
wc [options] [file ...]
```

- **Examples**:
    1. Count lines, words, and characters in a file:

```
wc file.txt
```

2. Display only the number of lines in a file:

```
wc -l file.txt
```

8. Diff

Definition:

The `diff` command is used to compare files line by line.

Usage/Formula

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

- **Examples**:

1. Compare two files and display the differences:

    ```
    diff file1.txt file2.txt
    ```

9. Grep

Definition :

The `grep` command is used to search for patterns in files.

Usage/Formula:

```
grep [options] pattern [file ...]
```

- **Examples**:

1. Search for a word in a file and display matching lines:

```
grep "pattern" file.txt
```

2. Search for a word recursively in all files within a directory:

```
grep -r "pattern" directory/
```