Ques2\_1.c:

* This implements the "word" command to count words in a file.
* It takes command line arguments to specify operation (-n for count, -d for difference) and filename(s).
* For -n, it opens the file and calls Number\_of\_words() to count words.
* For -d, it opens two files, counts words in each, prints the difference.
* Number\_of\_words() loops through the file character by character, incrementing count whenever it hits a word separator.
* It handles invalid arguments by printing usage.

Key points:

* Parse command line arguments
* Open and read files
* Modular function to count words
* Implement logic for different operations
* Robust argument checking and usage

Ques2\_2.c:

* Implements a customized "dir" command.
* Parses command line args to determine operation (-r to remove, -v for verbose output) and directory name.
* Checks if directory exists. Removes if -r given.
* Uses stat() to get directory info. Outputs if -v.
* Creates directory with mkdir().
* Handles errors like permission issues.

Key points:

* Custom command with flags
* Parse arguments
* Check directory existence
* Remove/verbose functionality
* Create directory
* Robust error handling

Ques2\_3.c:

* Implements a "date" command to print timestamp of a file.
* Parses args to get operation (-d for default, -R for RFC format) and filename.
* Checks file existence with stat().
* For -d, prints time with ctime().
* For -R, uses strftime() to format RFC timestamp and prints.
* Handles invalid args with usage.

Key points:

* Custom command with options
* Get file metadata
* Format output based on option
* Use standard library time functions
* Print timestamps in requested formats
* Robust argument checking