**Lab 5 (grep commands)**

**a.txt** file contain

Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do.

pattern matching is a key technique.

pattern1 is found here.

This line includes pattern2.

Pattern3 is useful for searching.

pattern has been mentioned in this line.

Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do.

Success

pattern pattern pattern

Dear Sir and Madam, we would like to inform you.

This line has pepper and is also properstore for tests.

This line has no punctuation

This line ends with a dash-

This line does not.

Vexing and bizarre incidents were reported.

there is another example line.

Finding ferns and flowers.

First and foremost.

Find the fortune in failures.

Success is the sum of small efforts - repeated day in and day out.

Success is achieved by those with discipline and dedication.

Sir and Madam should both be addressed.

rav is not here.

persistence ore, explore and adore.

Another line to test character limits beyond 40 characters indeed. Just another line.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

What a wonderful world of wonder and wisdom.

this Line has an Uppercase letter but not at the start.

pattern1, pattern2, and pattern3 should match with alternation.

there are empty lines above and below.

Certainly! I'll walk you through each grep command and script line-by-line, explaining how it works with a.txt as the file.

**Grep Commands with a.txt - Line-by-Line Explanation**

1. **Print all lines containing the word “pattern”**:

Copy code

**grep "pattern" a.txt**

* + grep "pattern" a.txt: Searches the file a.txt for any lines that contain the word "pattern".
  + **Explanation**: grep looks for the exact text "pattern" in each line of a.txt and prints matching lines.

**Pick out blank lines in the file:**

Copy code

**grep -E '^$' a.txt**

* + grep -E: Enables extended regular expressions.
  + '^$': Matches lines that start (^) and immediately end ($), with nothing in between (indicating an empty line).
  + **Explanation**: This finds and prints lines that are entirely blank.

1. **Count total number of empty lines in the file**:

Copy code

**grep -c '^$' a.txt**

* + -c: Instead of displaying each match, -c returns the count of matching lines.
  + **Explanation**: Counts the number of empty lines in a.txt.

1. **Print lines that contain both “Sir” and “Madam”**:

Copy code

**grep -E "Sir.\*Madam|Madam.\*Sir" a.txt**

* + "Sir.\*Madam": Searches for "Sir" followed by any characters (.\*) and then "Madam".
  + "Madam.\*Sir": Matches the reverse order (Madam first, then Sir).
  + |: Acts as an "OR" to match either pattern.
  + **Explanation**: Finds lines with both "Sir" and "Madam" in any order.

1. **Pick out lines with “pattern1”, “pattern2”, or “pattern3”**:

Copy code

**grep -E "pattern1|pattern2|pattern3" a.txt**

* + |: Alternates between matching "pattern1", "pattern2", or "pattern3".
  + **Explanation**: Prints lines that contain any of the three specified patterns.

1. **Lines with at least two ‘p’s followed by letters and ending with ‘ore’**:

Copy code

**grep -E "p.\*p.\*ore" a.txt**

* + "p.\*p.\*ore": Matches two occurrences of "p" separated by any characters, followed by "ore".
  + **Explanation**: This pattern searches for lines where there are at least two "p" characters and ending in "ore".

1. **Lines with “v”, “z”, or “I”**:

Copy code

**grep -E "[vzi]" a.txt**

* + [vzi]: A character class that matches any line containing “v”, “z”, or “I”.
  + **Explanation**: Finds lines that include any of these letters.

1. **Lines that do not start with an uppercase letter**:

Copy code

**grep -E "^[^A-Z]" a.txt**

* + "^[^A-Z]": The caret (^) matches the start of a line, and [^A-Z] matches any character that is not an uppercase letter.
  + **Explanation**: This finds lines that begin with a lowercase letter or other character.

1. **Lines ending with a dash (-)**:

Copy code

**grep -E "\-$" a.txt**

* + \-$: Escapes the dash symbol (-) so it's taken literally, and $ matches the end of the line.
  + **Explanation**: Finds lines that end with a dash.

1. **Words ending with “ore”**:

Copy code

**grep -o "\b\w\*ore\b" a.txt**

* -o: Only the matched parts are displayed.
* \b\w\*ore\b: Matches words ending in "ore", where \b represents word boundaries and \w\* represents any preceding letters.
* **Explanation**: Displays words ending with "ore".

1. **Words starting with “f” or “F”**:

Copy code

**grep -o "\b[fF]\w\*" a.txt**

* \b: Marks the start of a word.
* [fF]: Matches lowercase "f" or uppercase "F".
* \w\*: Matches any following word characters.
* **Explanation**: Finds words that start with "f" or "F".

1. **Lines with first-letter alliteration**:

Copy code

**grep -E "\b(\w)\w\*\s+\1\w\*" a.txt**

* (\w): Captures the first letter of a word.
* \s+: Requires a space between words.
* \1: Matches the same letter captured by (\w).
* **Explanation**: Finds lines with two adjacent words starting with the same letter.

1. **Count occurrences of the word “pattern”**:

Copy code

**grep -o "pattern" a.txt | wc -l**

* grep -o "pattern": Displays only the matched words.
* | wc -l: Pipes the output to wc -l, which counts the lines (i.e., occurrences).
* **Explanation**: Counts how often "pattern" appears.

1. **Lines with at least 40 characters**:

Copy code

**grep -E '^.{40,}$' a.txt**

* ^.{40,}$: Matches lines with any characters (.) of at least 40 in length.
* **Explanation**: Finds lines with at least 40 characters.

1. **Lines with no punctuation**:

Copy code

**grep -E '^[^[:punct:]]+$' a.txt**

* ^[^[:punct:]]+$: Matches lines containing no punctuation characters.
* **Explanation**: Finds lines with only letters or numbers, no punctuation.

1. **Lines with uppercase letters not at the start**:

Copy code

**grep -E '^.\\*[A-Z]' a.txt**

* ^.\\*[A-Z]: Matches lines with uppercase letters anywhere other than the first character.
* **Explanation**: Finds lines where an uppercase letter appears beyond the first position.

1. **Lines without “rav”**:

Copy code

**grep -v "rav" a.txt**

* -v: Inverts the search, excluding lines with "rav".
* **Explanation**: Finds lines that do not contain "rav".

**Shell Commands for File Manipulation - Line-by-Line Explanation**

1. **Script to count file types and print date**:

Copy code

**echo "Regular files: $(find . -type f | wc -l)"**

**echo "Links: $(find . -type l | wc -l)"**

**echo "Directories: $(find . -type d | wc -l)"**

**echo "Processed on: $(date)"**

* find . -type f | wc -l: Finds all regular files in the current directory.
* find . -type l | wc -l: Counts symbolic links.
* find . -type d | wc -l: Counts directories.
* $(date): Prints the current date.

1. **List directory contents to a.txt**:

Copy code

**ls -l > ~/a.txt**

* ls -l: Lists files with detailed information.
* > ~/a.txt: Redirects the output to a.txt in the home directory.

1. **Sort a.txt and append to contents-sorted.txt**:

Copy code

**sort ~/a.txt >> ~/contents-sorted.txt**

* sort ~/a.txt: Sorts the contents.
* >> ~/contents-sorted.txt: Appends sorted content to contents-sorted.txt.

1. **Redirect last 10 lines of /etc/passwd**:

Copy code

**tail -n 10 /etc/passwd > ~/Documents/passwd\_last\_10.txt**

* tail -n 10 /etc/passwd: Displays the last 10 lines of /etc/passwd.
* > ~/Documents/passwd\_last\_10.txt: Redirects output to passwd\_last\_10.txt.

1. **Count words in a.txt and append to field2.txt**:

Copy code

**wc -w < ~/a.txt >> ~/field2.txt**

* wc -w < ~/a.txt: Counts words in a.txt.
* >> ~/field2.txt: Appends the count to field2.txt.

1. **First 5 lines of /etc/passwd sorted in reverse**:

Copy code

**head -n 5 /etc/passwd | sort -r**

* head -n 5 /etc/passwd: Gets the first 5 lines.
* | sort -r: Sorts them in reverse alphabetical order.

1. **Count characters in last 9 lines of a.txt**:

Copy code

**tail -n 9 ~/a.txt | wc -m**

* tail -n 9 ~/a.txt: Retrieves the last 9 lines.
* | wc -m: Counts characters in those lines.