sed Commands

1. **Print the 5th line of a file using `sed`.**
2. **Replace the first occurrence of `foo` with `bar` in a file.**
3. **Replace all occurrences of `foo` with `bar` in a file.**
4. **Delete the 3rd line of a file.**
5. **Delete all lines containing the word `error`.**
6. **Insert the line `New line` after the 2nd line.**
7. **Append the line `End of file` at the end of the file.**
8. **Replace `None` with `Empty` globally in a file.**
9. **Print lines 10 to 20 of a file.**
10. **Print lines that do not contain `foo`.**
11. **Delete all empty lines from a file.**
12. **Replace `foo` with `bar` only if it appears at the beginning of a line.**
13. **Replace `foo` with `bar` only if it appears at the end of a line.**
14. **Change the 4th line to `Changed line`.**
15. **Add `Prefix:` to the beginning of each line.**
16. **Remove all leading spaces from each line.**
17. **Remove all trailing spaces from each line.**
18. **Remove lines containing only whitespace characters.**
19. **Replace `foo` with `bar` in a file and write changes to a new file.**
20. **Print lines that contain both `foo` and `bar`.**

awk Commands

21. **Print the 2nd field of each line from a file (assuming space delimiter).**
22. **Print the last field of each line from a file (assuming space delimiter).**
23. **Print lines where the 3rd field is greater than 50.**
24. **Sum the values in the 2nd field of a file.**
25. **Count the number of lines in a file.**

26. **Print lines where the 2nd field is equal to `foo`.**
27. **Replace the 1st field with `bar` in a file.**
28. **Print lines with more than 3 fields.**
29. **Print lines where the length of the 2nd field is greater than 5 characters.**
30. **Print the average of the values in the 4th field.**
31. **Print lines where the 1st field starts with `A`.**
32. **Print lines where the 2nd field ends with `X`.**
33. **Print the number of fields in each line.**
34. **Print lines that contain `foo` in the 2nd field.**
35. **Print the total number of characters in the 3rd field.**
36. **Print lines where the 1st field is a number greater than 10.**
37. **Add a new field with the value `new_field` to each line.**
38. **Print the 1st and 3rd fields of each line.**
39. **Sort lines based on the 2nd field.**
40. **Print the unique values in the 1st field.**

grep Commands

41. **Find all lines containing the word `foo` in a file.**
42. **Find all lines not containing the word `foo` in a file.**
43. **Count the number of lines containing the word `foo`.**
44. **Find lines containing `foo` but not `bar`.**
45. **Find lines that start with `foo`.**
46. **Find lines that end with `bar`.**
47. **Find lines containing `foo` and `bar` (both words must be present).**
48. **Find lines that contain exactly 5 characters.**
49. **Find lines that do not contain the word `error` and print line numbers.**
50. **Find lines containing `foo` ignoring case.**
51. **Find all occurrences of `foo` and show the line number.**
52. **Find lines containing `foo` and print only the matched text.**
53. **Print lines with 3 or more words.**

54. **Print lines where the 2nd word is `foo`.**

55. **Find all lines containing digits.**

56. **Find lines that are not empty.**

57. **Find lines where the 1st word is `foo`.**

58. **Search for lines containing a pattern from a file and print the file name.**

59. **Find lines containing `foo` and `bar`, where `foo` appears before `bar`.**

60. **Find lines with more than 80 characters.**

Combination Commands

61. **Use `sed` to remove all blank lines from a file, and then use `awk` to print the first field of each line.**

62. **Use `awk` to calculate the sum of the 2nd field and then use `grep` to find lines where the sum is greater than 100.**

63. **Find lines with `grep` containing `foo`, replace `foo` with `bar` using `sed`, and save the output to a new file.**

64. **Print lines with `awk` where the 3rd field is not empty, then use `sed` to remove the 2nd field from those lines.**

65. **Use `grep` to find lines containing `foo`, and then use `awk` to print the count of such lines.**

66. **Use `sed` to add a prefix `Line:` to each line, and then use `grep` to find lines containing `foo`.**

67. **Remove lines containing `None` using `sed`, and then use `awk` to print the 2nd field of each remaining line.**

68. **Find lines containing `foo` with `grep`, and then use `sed` to replace `foo` with `bar`.**

69. **Print the last field of lines where the 2nd field contains `foo` using `awk`, and then remove lines containing `foo` with `sed`.**

70. **Find all lines with `grep`, and use `awk` to print lines where the 1st field is greater than 10.**

Advanced Commands

71. **Use `awk` to print lines where the 1st and 2nd fields are equal, then use `sed` to replace `equal` with `match`.**

72. **Use `grep` to find lines with `foo`, and then use `awk` to count occurrences of `bar` in those lines.**

73. **Find lines containing `foo` using `grep`, and then use `sed` to delete lines with `bar`.**

74. **Use `awk` to print lines where the number of fields is greater than 3, then use `grep` to find lines containing `baz`.**

75. **Use `sed` to insert `Header:` at the beginning of the file, then use `awk` to print lines containing `foo`.**

76. **Count occurrences of `foo` in each line using `awk`, and then use `grep` to find lines where the count is greater than 2.**

77. **Find lines with more than 3 fields using `awk`, and then use `sed` to replace `foo` with `bar` in those lines.**

78. **Print lines where the 1st field starts with `A` using `awk`, and then use `grep` to find lines containing `foo` in the output.**

79. **Use `grep` to find lines with numbers, and then use `awk` to calculate the average of those numbers in the 2nd field.**

80. **Remove lines with `None` using `sed`, and then use `awk` to print the 3rd field of the remaining lines.**

File Manipulation

81. **Use `sed` to replace `foo` with `bar` in `file1`, and then use `grep` to find `bar` in `file2`.**

82. **Append the line `This is a new line.` to `file1`, then use `awk` to print the last line.**

83. **Use `awk` to calculate the sum of the 2nd field in `file1`, and then use `sed` to replace `sum` with the calculated value in `file2`.**

84. **Print lines from `file1` that do not exist in `file2` using `grep`, and then use `sed` to remove those lines from `file3`.**

85. **Use `grep` to find lines with `foo` in `file1`, then use `awk` to print the lines and save to `file2`.**

86. **Use `sed` to delete lines from `file1` that contain `foo`, then use `awk` to print the 1st field of the remaining lines and save to `file2`.**

87. **Find and replace `foo` with `bar` in `file1` using `sed`, and then use `grep` to count occurrences of `bar` in `file2`.**

88. **Print lines with `foo` from `file1` using `grep`, and then use `awk` to print the number of lines.**

89. **Use `awk` to find lines where the 2nd field is greater than 10 in `file1`,