1. Python Program to obtain the line number in which given word is present
2. Count number of lines in a text file in Python
3. Python Program to remove lines starting with any prefix
4. Python Program to Eliminate repeated lines from a file
5. Python Program to read List of Dictionaries from File
6. Python – Append content of one text file to another
7. Python program to copy odd lines of one file to other
8. Python Program to merge two files into a third file
9. Python program to Reverse a single line of a text file
10. Python program to reverse the content of a file and store it in another file
11. Python Program to Reverse the Content of a File using Stack
12. Here are Python programs for each of the tasks you listed:
13. ### 1. Python Program to Check if an Email Address is Valid or Not
14. ```python
15. import re
16. def is\_valid\_email(email):
17. pattern = r'^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$'
18. return re.match(pattern, email) is not None
19. email = input("Enter an email address: ")
20. if is\_valid\_email(email):
21. print("Valid email address")
22. else:
23. print("Invalid email address")
24. ```
25. ### 2. Python Program to Find Files Having a Particular Extension Using RegEx
26. ```python
27. import os
28. import re
29. def find\_files\_with\_extension(directory, extension):
30. pattern = re.compile(rf'.\*\.{extension}$')
31. for root, dirs, files in os.walk(directory):
32. for file in files:
33. if pattern.match(file):
34. print(os.path.join(root, file))
35. directory = input("Enter directory path: ")
36. extension = input("Enter file extension: ")
37. find\_files\_with\_extension(directory, extension)
38. ```
39. ### 3. Python Program to Extract IP Address from a File
40. ```python
41. import re
42. def extract\_ip\_addresses(filename):
43. with open(filename, 'r') as file:
44. content = file.read()
45. ips = re.findall(r'\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b', content)
46. return ips
47. filename = input("Enter the file name: ")
48. ip\_addresses = extract\_ip\_addresses(filename)
49. print("IP Addresses found:", ip\_addresses)
50. ```
51. ### 4. Python Program to Check the Validity of a Password
52. ```python
53. import re
54. def is\_valid\_password(password):
55. pattern = r'^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$'
56. return re.match(pattern, password) is not None
57. password = input("Enter a password: ")
58. if is\_valid\_password(password):
59. print("Valid password")
60. else:
61. print("Invalid password")
62. ```
63. ### 5. Categorize Password as Strong or Weak Using Regex in Python
64. ```python
65. import re
66. def categorize\_password(password):
67. strong\_pattern = r'^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$'
68. if re.match(strong\_pattern, password):
69. return "Strong"
70. else:
71. return "Weak"
72. password = input("Enter a password: ")
73. print("Password is:", categorize\_password(password))
74. ```
75. ### 6. Python Program to Read File Word by Word
76. ```python
77. def read\_file\_word\_by\_word(filename):
78. with open(filename, 'r') as file:
79. for line in file:
80. for word in line.split():
81. print(word)
82. filename = input("Enter the file name: ")
83. read\_file\_word\_by\_word(filename)
84. ```
85. ### 7. Python Program to Read Character by Character from a File
86. ```python
87. def read\_file\_character\_by\_character(filename):
88. with open(filename, 'r') as file:
89. while True:
90. char = file.read(1)
91. if not char:
92. break
93. print(char, end='')
94. filename = input("Enter the file name: ")
95. read\_file\_character\_by\_character(filename)
96. ```
97. ### 8. Python Program to Get Number of Characters, Words, Spaces, and Lines in a File
98. ```python
99. def file\_statistics(filename):
100. with open(filename, 'r') as file:
101. content = file.read()
102. num\_chars = len(content)
103. num\_words = len(content.split())
104. num\_lines = len(content.splitlines())
105. num\_spaces = content.count(' ')
107. return num\_chars, num\_words, num\_lines, num\_spaces
108. filename = input("Enter the file name: ")
109. chars, words, lines, spaces = file\_statistics(filename)
110. print(f"Characters: {chars}, Words: {words}, Lines: {lines}, Spaces: {spaces}")
111. ```
112. ### 9. Python Program to Count the Number of Occurrences of a Key-Value Pair in a Text File
113. ```python
114. def count\_key\_value\_occurrences(filename, key, value):
115. count = 0
116. with open(filename, 'r') as file:
117. for line in file:
118. if f"{key}: {value}" in line:
119. count += 1
120. return count
121. filename = input("Enter the file name: ")
122. key = input("Enter the key: ")
123. value = input("Enter the value: ")
124. occurrences = count\_key\_value\_occurrences(filename, key, value)
125. print(f"Occurrences of {key}: {value} = {occurrences}")
126. ```
127. ### 10. Python Program to Find ‘n’ Character Words in a Text File
128. ```python
129. import re
130. def find\_n\_character\_words(filename, n):
131. with open(filename, 'r') as file:
132. content = file.read()
133. pattern = rf'\b\w{{{n}}}\b'
134. words = re.findall(pattern, content)
135. return words
136. filename = input("Enter the file name: ")
137. n = int(input("Enter the number of characters: "))
138. words = find\_n\_character\_words(filename, n)
139. print(f"Words with {n} characters:", words)
140. ```
141. These programs use Python's standard libraries such as `os` for file operations and `re` for regular expression matching to accomplish the tasks.