1. Python – Replace multiple words with K
2. Python | Permutation of a given string using inbuilt function
3. Python | Check for URL in a String
4. Execute a String of Code in Python
5. String slicing in Python to rotate a string
6. String slicing in Python to check if a string can become empty by recursive deletion
7. Python Counter| Find all duplicate characters in string
8. Python – Replace all occurrences of a substring in a string
9. Python – Extract Unique values dictionary values
10. Python program to find the sum of all items in a dictionary
11. Here's how you can approach each of these tasks in Python:
12. 1. \*\*Replace multiple words with 'K':\*\*
13. ```python
14. def replace\_words(text, words\_to\_replace):
15. for word in words\_to\_replace:
16. text = text.replace(word, 'K')
17. return text
18. text = "Python is great and Python is fun"
19. words\_to\_replace = ['Python', 'great']
20. result = replace\_words(text, words\_to\_replace)
21. print(result) # Output: K is K and K is fun
22. ```
23. 2. \*\*Permutation of a given string using an inbuilt function:\*\*
24. ```python
25. from itertools import permutations
26. def get\_permutations(string):
27. return [''.join(p) for p in permutations(string)]
28. string = "ABC"
29. result = get\_permutations(string)
30. print(result) # Output: ['ABC', 'ACB', 'BAC', 'BCA', 'CAB', 'CBA']
31. ```
32. 3. \*\*Check for URL in a string:\*\*
33. ```python
34. import re
35. def check\_for\_url(text):
36. url\_pattern = re.compile(r'(https?://[^\s]+)')
37. return re.findall(url\_pattern, text)
38. text = "Check out https://example.com and http://example.org"
39. result = check\_for\_url(text)
40. print(result) # Output: ['https://example.com', 'http://example.org']
41. ```
42. 4. \*\*Execute a string of code in Python:\*\*
43. ```python
44. code = "print('Hello, World!')"
45. exec(code) # Output: Hello, World!
46. ```
47. 5. \*\*String slicing in Python to rotate a string:\*\*
48. ```python
49. def rotate\_string(s, n):
50. return s[n:] + s[:n]
51. string = "GeeksforGeeks"
52. rotated\_string = rotate\_string(string, 3)
53. print(rotated\_string) # Output: ksforGeeksGee
54. ```
55. 6. \*\*String slicing in Python to check if a string can become empty by recursive deletion:\*\*
56. ```python
57. def can\_become\_empty(s, sub):
58. while sub in s:
59. s = s.replace(sub, '')
60. return len(s) == 0
61. s = "GEEGEEKSSFORGEEKS"
62. sub = "GEEKS"
63. result = can\_become\_empty(s, sub)
64. print(result) # Output: True
65. ```
66. 7. \*\*Python Counter | Find all duplicate characters in a string:\*\*
67. ```python
68. from collections import Counter
69. def find\_duplicates(s):
70. count = Counter(s)
71. return [char for char, cnt in count.items() if cnt > 1]
72. string = "GeeksforGeeks"
73. duplicates = find\_duplicates(string)
74. print(duplicates) # Output: ['e', 'k', 's', 'G']
75. ```
76. 8. \*\*Replace all occurrences of a substring in a string:\*\*
77. ```python
78. def replace\_substring(s, old, new):
79. return s.replace(old, new)
80. string = "Hello World! World is beautiful."
81. result = replace\_substring(string, "World", "Earth")
82. print(result) # Output: Hello Earth! Earth is beautiful.
83. ```
84. 9. \*\*Extract unique values from dictionary values:\*\*
85. ```python
86. def extract\_unique\_values(dic):
87. unique\_values = set()
88. for value in dic.values():
89. unique\_values.update(value)
90. return list(unique\_values)
91. dic = {'a': [1, 2, 3], 'b': [2, 4, 6], 'c': [5, 7, 9]}
92. unique\_values = extract\_unique\_values(dic)
93. print(unique\_values) # Output: [1, 2, 3, 4, 5, 6, 7, 9]
94. ```
95. 10. \*\*Python program to find the sum of all items in a dictionary:\*\*
96. ```python
97. def sum\_of\_dict\_items(dic):
98. return sum(dic.values())
99. dic = {'a': 100, 'b': 200, 'c': 300}
100. total\_sum = sum\_of\_dict\_items(dic)
101. print(total\_sum) # Output: 600
102. ```
103. These Python snippets provide solutions for the tasks you've listed. Let me know if you need further explanations or modifications!