1. Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700 (both included).
2. Write a Python program that accepts a word from the user and reverse it.
3. Write a Python program to calculate a dog's age in dog's years.  Note: For the first two years, a dog year is equal to 10.5 human years. After that, each dog year equals 4 human years.
4. Write a Python program to check a triangle is equilateral, isosceles or scalene
5. Write a Python program to detect the number of local variables declared in a function.
6. Write a Python program to execute a string containing Python code.
7. Write a Python program to convert a given list of tuples to a list of strings using map function.
8. Write a Python program to split a given dictionary of lists into list of dictionaries using map function.
9. Write a Python program to scan a specified directory and identify the sub directories and files.
10. Write a Python program to parse a string representing time and returns the structure time.
11. Write a Python code to send some sort of data in the URL's query string.
12. Write a Python code to send a request to a web page, and print the information of headers. Also parse these values and print key-value pairs holding various information.
13. Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.
14. Write a Python program that reads each row of a given csv file and skip the header of the file. Also print the number of rows and the field names.
15. Write a Python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9)  
    Here are the Python programs for each of the tasks:
16. 1. \*\*Find numbers divisible by 7 and multiple of 5:\*\*
17. ```python
18. numbers = []
19. for i in range(1500, 2701):
20. if i % 7 == 0 and i % 5 == 0:
21. numbers.append(i)
22. print(numbers)
23. ```
24. 2. \*\*Reverse a word:\*\*
25. ```python
26. word = input("Enter a word: ")
27. reversed\_word = word[::-1]
28. print("Reversed word:", reversed\_word)
29. ```
30. 3. \*\*Calculate a dog's age in dog years:\*\*
31. ```python
32. human\_years = int(input("Input a dog's age in human years: "))
33. if human\_years <= 2:
34. dog\_years = human\_years \* 10.5
35. else:
36. dog\_years = 21 + (human\_years - 2) \* 4
37. print("The dog's age in dog years is:", dog\_years)
38. ```
39. 4. \*\*Check if a triangle is equilateral, isosceles, or scalene:\*\*
40. ```python
41. a = float(input("Enter first side: "))
42. b = float(input("Enter second side: "))
43. c = float(input("Enter third side: "))
44. if a == b == c:
45. print("The triangle is equilateral.")
46. elif a == b or b == c or c == a:
47. print("The triangle is isosceles.")
48. else:
49. print("The triangle is scalene.")
50. ```
51. 5. \*\*Detect the number of local variables declared in a function:\*\*
52. ```python
53. def sample\_function():
54. x = 1
55. y = 2
56. z = 3
57. print("Number of local variables:", sample\_function.\_\_code\_\_.co\_nlocals)
58. ```
59. 6. \*\*Execute a string containing Python code:\*\*
60. ```python
61. code = "print('Hello, World!')"
62. exec(code)
63. ```
64. 7. \*\*Convert a list of tuples to a list of strings using `map`:\*\*
65. ```python
66. tuples\_list = [(1, 2), (3, 4), (5, 6)]
67. strings\_list = list(map(str, tuples\_list))
68. print(strings\_list)
69. ```
70. 8. \*\*Split a dictionary of lists into a list of dictionaries using `map`:\*\*
71. ```python
72. dict\_of\_lists = {'a': [1, 2], 'b': [3, 4], 'c': [5, 6]}
73. list\_of\_dicts = list(map(dict, zip(\*[[(key, val) for val in value] for key, value in dict\_of\_lists.items()])))
74. print(list\_of\_dicts)
75. ```
76. 9. \*\*Scan a directory and identify subdirectories and files:\*\*
77. ```python
78. import os
79. directory = "your\_directory\_path"
80. for root, dirs, files in os.walk(directory):
81. print("Subdirectories:", dirs)
82. print("Files:", files)
83. ```
84. 10. \*\*Parse a string representing time and return a structured time:\*\*
85. ```python
86. import time
87. time\_string = "21 June, 2021"
88. parsed\_time = time.strptime(time\_string, "%d %B, %Y")
89. print(parsed\_time)
90. ```
91. 11. \*\*Send data in URL's query string:\*\*
92. ```python
93. import urllib.parse
94. base\_url = "https://example.com/"
95. params = {'name': 'John', 'age': '25'}
96. query\_string = urllib.parse.urlencode(params)
97. full\_url = base\_url + "?" + query\_string
98. print("URL with query string:", full\_url)
99. ```
100. 12. \*\*Send a request to a web page and print the headers:\*\*
101. ```python
102. import requests
103. url = 'https://example.com'
104. response = requests.get(url)
105. print("Headers:")
106. for key, value in response.headers.items():
107. print(f"{key}: {value}")
108. ```
109. 13. \*\*Write a Python dictionary to a CSV file and read it back:\*\*
110. ```python
111. import csv
112. data = {'Name': 'John', 'Age': 25, 'City': 'New York'}
113. with open('output.csv', 'w', newline='') as file:
114. writer = csv.DictWriter(file, fieldnames=data.keys())
115. writer.writeheader()
116. writer.writerow(data)
117. with open('output.csv', newline='') as file:
118. reader = csv.reader(file)
119. for row in reader:
120. print(row)
121. ```
122. 14. \*\*Read a CSV file, skip the header, and print rows and field names:\*\*
123. ```python
124. import csv
125. with open('input.csv', newline='') as csvfile:
126. reader = csv.reader(csvfile)
127. field\_names = next(reader)
128. print("Field names:", field\_names)
129. row\_count = 0
130. for row in reader:
131. row\_count += 1
132. print(row)
133. print("Total number of rows:", row\_count)
134. ```
135. 15. \*\*Check that a string contains only a certain set of characters:\*\*
136. ```python
137. import re
138. string = input("Enter a string: ")
139. pattern = re.compile("^[a-zA-Z0-9]\*$")
140. if pattern.match(string):
141. print("String contains only the specified characters.")
142. else:
143. print("String contains characters outside the specified set.")
144. ```
145. These programs address the tasks you've outlined. Feel free to modify them as needed!