1. # utils/gemini\_client.py
2. import os
3. import google.generativeai as genai
4. from dotenv import load\_dotenv
5. load\_dotenv()
6. API\_KEY = os.getenv("GEMINI\_API\_KEY")
7. if not API\_KEY:
8. raise ValueError("GEMINI\_API\_KEY not found in environment

variables.")

1. genai.configure(api\_key=API\_KEY)
2. model = genai.GenerativeModel("gemini-2.0-flash")
3. def generate\_answer(prompt: str) -> str:
4. try:
5. response = model.generate\_content(
6. prompt,
7. generation\_config={"temperature": 0.7}
8. )
9. return response.text.strip()
10. except Exception as e:
11. return f"[ERROR] {str(e)}"
12. # utils/io\_utils.py
14. import json
15. def load\_json(path: str):
16. with open(path, "r", encoding="utf-8") as f:
17. return json.load(f)
18. def save\_json(data, path: str):
19. with open(path, "w", encoding="utf-8") as f:
20. json.dump(data, f, indent=2, ensure\_ascii=False)
21. # main.py
22. from utils.io\_utils import load\_json, save\_json
23. from utils.gemini\_client import generate\_answer
24. import re
25. import os
26. import argparse
27. import pandas as pd
28. import time
29. def parse\_args():
30. parser = argparse.ArgumentParser(description="Run prompting with

Gemini API.")

1. parser.add\_argument(

        "--output",

        type=str,

        default="generate\_res/dummy.json",

        help="Output file path (default: generate\_res/dummy.json)"

    )

1. parser.add\_argument(

        "--task",

        type=str,

        default="commaqa",

        help="Dataset type ? (default: commaqa)"

    )

1. return parser.parse\_args()
2. def get\_controller(task):
3. if task == "reverse":
4. from reverse\_controller import ReverseController
5. return ReverseController()
6. elif task == "lettercat":
7. from lettercat\_controller import LatterCatController
8. return LatterCatController()
9. elif task == "commaqa":
10. from commaqa\_controller import CommAQAController
11. return CommAQAController()
12. else:
13. raise ValueError(f"Unsupported task: {task}")
14. def main():
15. args = parse\_args()
16. print("Arguments parsed:")
17. for arg, value in vars(args).items():
18. print(f"{arg}: {value}")
19. print("\n")
20. controller = get\_controller(args.task)
21. # get decomposition chains
22. controller.solve()
24. if \_\_name\_\_ == "\_\_main\_\_":
25. main()
26. # reverse\_controller.py
27. from utils.io\_utils import load\_json, save\_json
28. import prompts.reverse.template as reverse\_template
29. from utils.gemini\_client import generate\_answer
30. import re
31. import os
32. import time
33. class ReverseController:
34. def \_\_init\_\_(self):
35. self.dataset = load\_json('dataset/reverse.json')
36. def print\_questions(self):
37. """Prints all questions from the dataset."""
38. for entry in self.dataset:
39. print(entry['question'])
40. def replace\_references(self, s, result\_array):
41. def replacer(match):
42. index = int(match.group(1))-1  # Ambil angka dari #3 misalnya
43. if 0 <= index < len(result\_array):
44. return result\_array[index]
45. else:
46. return f"<Invalid index #{index}>"
48. # Ganti semua #angka dengan nilai yang sesuai
49. return re.sub(r"#(\d+)", replacer, s)
50. def extract\_qs(self, text):
51. matches = re.findall(r'.\* (?:is|reverse) \"(.\*)\"\.', text)
52. return matches[0] if matches else ""
53. def get\_generated\_answer(self, ans):
54. if "Answer:" in ans:
55. process\_answer = ans.split("Answer: ", 1)[1].strip()
56. else:
57. process\_answer = ans.strip()
58. return process\_answer
59. def chain\_processing(self, qs\_lines):
60. result\_answers = []
61. for index, qs in enumerate(qs\_lines):
62. match = re.match(r"\[(\w+)]", qs)
63. match = match.group(1)
64. process\_answer = ""
65. if match == "extract":
66. process\_answer = self.extract\_qs(qs)
67. elif match == "remove\_numbers":
68. formatted\_qs = self.replace\_references(qs, result\_answers)
69. rm\_num\_prompt = reverse\_template.remove\_numbers\_template

.replace('{input}', formatted\_qs)

1. generated\_ans = generate\_answer(rm\_num\_prompt)
2. print(f"Generated answer for remove\_numbers: {generated\_ans}")
3. process\_answer = self.get\_generated\_answer(generated\_ans)
5. elif match == "reverse":
6. formatted\_qs = self.replace\_references(qs, result\_answers)
7. cot\_prompt = reverse\_template.cot\_template.replace('{input}',

formatted\_qs)

1. print(f"COT prompt for reverse: {cot\_prompt}")
2. generated\_ans = generate\_answer(cot\_prompt)
3. print(f"Generated answer for reverse: {generated\_ans}")
4. process\_answer = self.get\_generated\_answer(generated\_ans)
6. elif match == "join":
7. formatted\_qs = self.replace\_references(qs, result\_answers)
8. join\_prompt = reverse\_template.join\_template.replace('{input}',

formatted\_qs)

1. generated\_ans = generate\_answer(join\_prompt)
2. process\_answer = self.get\_generated\_answer(generated\_ans)
4. elif match == "EOQ":
5. break
7. print(f"Processing step {index+1}: {match} -> {process\_answer}")
8. result\_answers.append(process\_answer)
9. return result\_answers
10. def generate\_chain(self, question):
11. template = reverse\_template.decomp\_chain
12. prompt = template.replace('{input}', question)
13. decomp\_chain = generate\_answer(prompt)
15. qs\_lines = [line.replace("QS: ", "").strip() for line in

decomp\_chain.splitlines() if line.startswith("QS:")]

1. print(f"Decomposition chain generated: {qs\_lines}")
2. return qs\_lines
4. def solve(self):
5. output\_log = []
6. for entry in self.dataset:
8. qs\_lines = self.generate\_chain(entry['question'])
9. output = self.chain\_processing(qs\_lines)
11. output\_log.append({
12. 'question': entry['question'],
13. 'qs\_lines': qs\_lines,
14. 'result': output,
15. "is\_correct": entry['answer']["spans"][0] == output[-1] if

output else False

1. })
2. save\_json(output\_log, 'generate\_res/dummy.json')
3. time.sleep(60)  # To avoid hitting rate limits
4. return True
5. # lettercat\_controller.py
6. from utils.io\_utils import load\_json, save\_json
7. import prompts.lettercat.template as lettercat\_template
8. from utils.gemini\_client import generate\_answer
9. import re
10. import os
11. import time
12. import json
13. class LatterCatController:
14. def \_\_init\_\_(self):
15. self.dataset = load\_json('dataset/lettercat.json')
16. def print\_questions(self):
17. """Prints all questions from the dataset."""
18. for entry in self.dataset:
19. print(entry['question'])
20. def replace\_references(self, s, result\_array):
21. def replacer(match):
22. index = int(match.group(1))-1  # Ambil angka dari #3 misalnya
23. if 0 <= index < len(result\_array):
24. return result\_array[index]
25. else:
26. return f"<Invalid index #{index}>"
28. # Ganti semua #angka dengan nilai yang sesuai
29. return re.sub(r"#(\d+)", replacer, s)
31. def get\_references(self, qs):
32. references = re.findall(r'#(\d+)', qs)
33. return [int(ref) for ref in references]
34. def get\_generated\_answer(self, ans):
35. if "Answer:" in ans:
36. process\_answer = ans.split("Answer: ", 1)[1].strip()
37. else:
38. process\_answer = ans.strip()
39. return process\_answer
40. def chain\_processing(self, qs\_lines):
41. result\_answers = []
42. for index, qs in enumerate(qs\_lines):
43. match = re.search(r'\[([^\]]+)\]', qs)
44. match = match.group(1)
45. process\_answer = ""
46. if match == "split":
47. split\_prompt = lettercat\_template.split\_template.replace('{input}',

qs)

1. generated\_ans = generate\_answer(split\_prompt)
2. print(f"Generated answer for split: {generated\_ans}")
3. process\_answer = self.get\_generated\_answer(generated\_ans)
5. elif match == "str\_position":
6. references = self.get\_references(qs)
7. word\_lists = json.loads(result\_answers[references[0]-1])
8. ans\_list = []
9. time.sleep(60)
10. for word in word\_lists:
11. formatted\_qs = re.sub(r"#\d+", word, qs)
12. print("Formatted qs: ")
13. print(formatted\_qs)
14. cleaned\_formatted\_qs = re.sub(r'\(.\*?\)|\[.\*?\]', '',

formatted\_qs).strip()

1. print("Cleaned formatted qs: ")
2. print(cleaned\_formatted\_qs)
3. str\_pos\_decomp\_chain = self.generate\_chain(lettercat\_template

.str\_position\_template, cleaned\_formatted\_qs)

1. ans\_list.append(self.chain\_processing(str\_pos\_decomp\_chain)[-

1])

1. print(f"Split Result: {ans\_list}")
2. print(ans\_list)
3. process\_answer = str(ans\_list)
4. elif match == "arr\_position":
5. formatted\_qs = self.replace\_references(qs, result\_answers)
6. arr\_pos\_prompt = lettercat\_template.arr\_position\_template

.replace('{input}', formatted\_qs)

1. generated\_ans = generate\_answer(arr\_pos\_prompt)
2. process\_answer = self.get\_generated\_answer(generated\_ans)
4. elif match == "merge":
5. formatted\_qs = self.replace\_references(qs, result\_answers)
6. print("Merge Process")
7. print(formatted\_qs)
8. merge\_prompt = lettercat\_template.merge\_template.replace('{input}',

formatted\_qs)

1. generated\_ans = generate\_answer(merge\_prompt)
2. print(f"Generated answer for merge: {generated\_ans}")
3. process\_answer = self.get\_generated\_answer(generated\_ans)
5. elif match == "EOQ":
6. break
8. # print(f"Processing step {index+1}: {match} -> {process\_answer}")
9. result\_answers.append(process\_answer)
10. return result\_answers
11. def generate\_chain(self, template, question):
12. prompt = template.replace('{input}', question)
13. decomp\_chain = generate\_answer(prompt)
15. qs\_lines = [line.replace("QS: ", "").strip() for line in

decomp\_chain.splitlines() if line.startswith("QS:")]

1. return qs\_lines
3. def solve(self):
4. output\_log = []
5. for entry in self.dataset:
7. print(f"Processing question: {entry['question']}")
8. template = lettercat\_template.decomp\_template
9. qs\_lines = self.generate\_chain(template, entry['question'])
10. output = self.chain\_processing(qs\_lines)
12. output\_log.append({
13. 'question': entry['question'],
14. 'qs\_lines': qs\_lines,
15. 'result': output,
16. "is\_correct": entry['answer']["spans"][0] == output[-1] if

output else False

1. })
2. save\_json(output\_log, 'generate\_res/dummy.json')
3. time.sleep(60)  # To avoid hitting rate limits
4. return True
5. # commaqa\_controller.py
6. from utils.io\_utils import load\_json, save\_json
7. import prompts.commaQA.template as commaQA\_template
8. from utils.gemini\_client import generate\_answer
9. import re
10. import os
11. import time
12. import json
13. import ast
14. class CommAQAController:
15. def \_\_init\_\_(self):
16. self.dataset = load\_json('dataset/commaqa.json')
17. def print\_questions(self):
18. """Prints all questions from the dataset."""
19. for entry in self.dataset:
20. print(entry['question'])
21. def replace\_references(self, s, result\_array):
22. def replacer(match):
23. index = int(match.group(1))-1  # Ambil angka dari #3 misalnya
24. if 0 <= index < len(result\_array):
25. return result\_array[index]
26. else:
27. return f"<Invalid index #{index}>"
29. # Ganti semua #angka dengan nilai yang sesuai
30. return re.sub(r"#(\d+)", replacer, s)
32. def get\_references(self, qs):
33. references = re.findall(r'#(\d+)', qs)
34. return [int(ref) for ref in references]
36. def get\_generated\_answer(self, ans):
37. if "Answer:" in ans:
38. process\_answer = ans.split("Answer: ", 1)[1].strip()
39. else:
40. process\_answer = ans.strip()
41. return process\_answer
43. def get\_Qproject\_values\_flat\_unique(self, result\_array):
44. return
46. def clean\_text(self, text):
47. return re.sub(r'\(.\*?\)|\[.\*?\]', '', text).strip()
48. def get\_references\_array(self, qs, result\_answers):
49. references = result\_answers[self.get\_references(qs)[0]-1]
50. print(f"references: {references}")
51. array\_ver = json.loads(references)
52. print(array\_ver)
53. return array\_ver
54. def get\_score(self, output, entry\_answer):
55. ans = ast.literal\_eval(output[-1])
56. ans\_expected = entry\_answer["spans"]
57. total\_number = len(ans\_expected)
58. temp\_score = 0
59. for ans\_exp in ans\_expected:
60. if ans\_exp in ans:
61. temp\_score += 1
62. return temp\_score/total\_number
63. def simple\_subtask(self, qs, prompt\_template, passage=""):
64. cleaned\_text = self.clean\_text(qs)
65. prompt = prompt\_template.replace('{input}', cleaned\_text)

.replace('{passage}', passage)

1. generated\_ans = generate\_answer(prompt)
2. return self.get\_generated\_answer(generated\_ans)
4. def project\_values\_flat\_unique\_subtask(self, qs, result\_answers,

prompt\_template, passage=""):

1. references = self.get\_references\_array(qs, result\_answers)
2. ans\_list = []
3. print(f"QA Prompt References: {references}")
4. for ref in references:
5. # format and generate prompt template
6. formatted\_qs = re.sub(r"#\d+", ref, qs)
7. cleaned\_text = self.clean\_text(formatted\_qs)
8. prompt = prompt\_template.replace('{input}', cleaned\_text)

.replace('{passage}', passage)

1. # generate answer
2. generated\_ans = generate\_answer(prompt)
3. ans\_list.append(json.loads(self.get\_generated\_answer(generated\_ans)))
4. # convert to SET DataStructure add to process answer
5. ans\_set\_list = list(set(item for sublist in ans\_list for item in sublist))
6. return json.dumps(ans\_set\_list)
7. def chain\_processing(self, qs\_lines, passage=""):
8. result\_answers = []
9. for index, qs in enumerate(qs\_lines):
10. # Cari yang dalam tanda kurung bulat (command)
11. command\_match = re.search(r"\[(.\*?)\]", qs)
12. command = command\_match.group(1) if command\_match else ''
14. # Cari yang dalam tanda kurung kotak (type)
15. type\_match = re.search(r"\((.\*?)\)", qs)
16. query\_type = type\_match.group(1) if type\_match else ''
17. process\_answer = ""
18. if command == "simp\_qa":
19. if query\_type == "select":
20. process\_answer = self.simple\_subtask(qs, commaQA\_template

.simp\_qa\_template, passage)

2. elif query\_type == "project\_values\_flat\_unique":
3. process\_answer = self.project\_values\_flat\_unique\_subtask(qs,

result\_answers,

commaQA\_template.simp\_qa\_template, passage)

1. elif command == "pos\_qa":
2. if query\_type == "select":
3. process\_answer = self.simple\_subtask(qs,

commaQA\_template.pos\_qa\_template, passage)

1. elif query\_type == "project\_values\_flat\_unique":
2. process\_answer = self.project\_values\_flat\_unique\_subtask(qs,

result\_answers,

commaQA\_template.pos\_qa\_template, passage)

2. elif command == "aw\_qa":
3. if query\_type == "select":
4. process\_answer = self.simple\_subtask(qs,

commaQA\_template.aw\_qa\_template, passage)

1. elif query\_type == "project\_values\_flat\_unique":
2. process\_answer = self.project\_values\_flat\_unique\_subtask(qs,

result\_answers, commaQA\_template.aw\_qa\_template, passage)

1. elif command == "EOQ":
2. Break
4. print(f"Processing step {index+1}: {command} -> {process\_answer}")
5. result\_answers.append(process\_answer)
6. return result\_answers
7. def generate\_chain(self, question):
8. template = commaQA\_template.decomp\_qa\_template
9. prompt = template.replace('{input}', question)
10. decomp\_chain = generate\_answer(prompt)
12. qs\_lines = [line.replace("QS: ", "").strip() for line in

decomp\_chain.splitlines() if line.startswith("QS:")]

1. return qs\_lines
3. def solve(self):
4. output\_log = []
5. score = 0
6. for key, value in self.dataset.items():
7. for index, entry in enumerate(value.get('qa\_pairs')):
8. qs\_lines = self.generate\_chain(entry['question'])
9. print(f"PROCESS START - {key} - {index}")
10. print(f"lines: {qs\_lines}")
11. output = self.chain\_processing(qs\_lines,

passage=value.get('passage'))

2. current\_score = self.get\_score(output, entry['answer'])
3. output\_log.append({
4. 'passage': value.get('passage'),
5. 'question': entry['question'],
6. 'qs\_lines': qs\_lines,
7. 'result': output,
8. "score": current\_score
9. })
10. print(f"score : {current\_score}\n")
11. score += current\_score
12. save\_json(output\_log, 'generate\_res/dummy.json')
13. time.sleep(60)  # To avoid hitting rate limits
14. print(f"final score: {score}")
15. return True
16. # prompts/reverse/template.py
17. decomp\_chain = '''Generate a Question Set (QS) based on the given Question Context (QC) WITHOUT PROVIDING Answer (A). Each QS step should logically follow from the previous one, leading to the final answer, but do not include the answer. Use consistent format and tags as shown below.
18. QC: Reverse the sequence "driving license, button, packet, identity card, shoe".
19. QS: [extract] The sequence is "1. driving license, 2. button, 3. packet, 4. identity card, 5. shoe". The sequence is 5 items long, which is more than the minimum length of 4, so we split it. Half of 5 is 5 / 2 = 2.5. Dropping the decimal, we get that the first half will be 2 items long, ending in "2. button". The first half (2 items) is "1. driving license, 2. button".
20. QS: [extract] The first half of the sequence ends with "2. button", so the second half starts after "2. button" with "3. packet". The full sequence is 5 items long, and the first half is 2 items long, so the second half will be 5 - 2 = 3 items long. The second half of the sequence (3 items) is "3. packet, 4. identity card, 5. shoe".
21. QS: [remove\_numbers] Remove the numbers from #1.
22. QS: [remove\_numbers] Remove the numbers from #2.
23. QS: [reverse] Reverse the sequence #3.
24. QS: [reverse] Reverse the sequence #4.
25. QS: [join] #6 #5
26. QS: [EOQ]
27. QC: Reverse the sequence "newspaper, glasses, laptop, bottle".
28. QS: [extract] The sequence is "1. newspaper, 2. glasses, 3. laptop, 4. bottle". The sequence is 4 items long, which is equal to the minimum length of 4, so we split it. Half of 4 is 4 / 2 = 2.0. Dropping the decimal, we get that the first half will be 2 items long. The first half (2 items) of the sequence is "1. newspaper, 2. glasses".
29. QS: [extract] The first half of the sequence ends with "2. glasses", so the second half starts after "2. glasses" with "3. laptop". The full sequence is 4 items long and the first half is 2 items long, so the second half will be 4 - 2 = 2 items long, ending in "2. glasses". The second half of the sequence (2 items) is "3. laptop, 4. bottle".
30. QS: [remove\_numbers] Remove the numbers from #1.
31. QS: [remove\_numbers] Remove the numbers from #2.
32. QS: [reverse] Reverse the sequence #3.
33. QS: [reverse] Reverse the sequence #4.
34. QS: [join] #6 #5
35. QS: [EOQ]
36. QC: {input}
37. Output:
38. QS: <QS-1>
39. QS: <QS-2>
40. .....
41. QS: <QS-N>
42. '''
43. join\_template = '''
44. Q: "bottle, laptop" "glasses, newspaper".
45. Answer: bottle, laptop, glasses, newspaper
46. Q: "identity card, packet, button" "magazine, notebook, glasses".
47. Answer: identity card, packet, button, magazine, notebook, glasses
48. Q: "passport, umbrella, radio, mobile phone, photo" "player".
49. Answer:  passport, umbrella, radio, mobile phone, photo, player
50. Q: "mirror, case" "toothbrush, alarm clock".
51. Answer: mirror, case, toothbrush, alarm clock
52. Q: "light bulb, clip, umbrella" "driving licence, watch".
53. Answer: light bulb, clip, umbrella, driving licence, watch
54. Q: {input}
55. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
56. Answer: <your answer here>
57. '''
58. remove\_numbers\_template = '''
59. Q: Remove the numbers from "4. bottle, 3. laptop, 2. glasses, 1. newspaper".
60. Answer: bottle, laptop, glasses, newspaper
61. Q: Remove the numbers from "1. identity card, 2. packet, 3. button".
62. Answer: identity card, packet, button
63. Q: Remove the numbers from "1. player, 2. passport, 3. umbrella, 4. radio".
64. Answer: player, passport, umbrella, radio
65. Q: {input}
66. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
67. Answer: <your answer here>
68. '''
69. cot\_template = '''
70. QC: Reverse the sequence "newspaper, glasses, laptop, bottle".
71. QS: First is newspaper. Second is glasses. Third is laptop. Fourth is bottle. Now to reverse, change the order to: Fourth is bottle. Third is laptop. Second is glasses. First is newspaper. So the answer is "bottle, laptop, glasses, newspaper".
72. Answer: bottle, laptop, glasses, newspaper
73. QC: Reverse the sequence "laptop, photo, clip".
74. QS: First is laptop. Second is photo. Third is clip. Now to reverse, change the order to: Third is clip. Second is photo. First is laptop. So the answer is "clip, photo, laptop".
75. Answer: clip, photo, laptop
76. QC: Reverse the sequence "driving license, button, packet, identity card, pineapple".
77. QS: First is driving license. Second is button. Third is packet. Fourth is identity card. Fifth is pineapple. Now to reverse, change the order to: Fifth is pineapple. Fourth is identity card. Third is packet. Second is button. First is driving license. So the answer is "pineapple, identity card, packet, button, driving license".
78. Answer: pineapple, identity card, packet, button, driving license
79. Q: {nput}
80. QS: <your qs answer here>
81. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
82. Answer: <your final answer here>
83. '''
84. # prompts/lettercat/template.py
85. decomp\_template = '''
86. QC: Take the last letters of the words in "Elon Musk Tesla" and concatenate them.
87. QS: [split] What are the words in "Elon Musk"?
88. QS: (project\_values) [str\_position] What is the last letter in "#1"?
89. QS: [merge] Concatenate #2 using a space.
90. QS: [EOQ]
91. QC: Take the letters at position 1 of the words in "Sundar Pichai Google" and concatenate them.
92. QS: [split] What are the words in "Sundar Pichai Google"?
93. QS: (project\_values) [str\_position] What is the letter at position 1 in "#1"?
94. QS: [merge] Concatenate #2 using a space.
95. QS: [EOQ]
96. QC: Take the letters at position 4 of the words in "Oren Etzioni AllenInstitute" and concatenate them.
97. QS: [split] What are the words in "Oren Etzioni AllenInstitute"?
98. QS: (project\_values) [str\_position] What is the letter at position 4 in "#1"?
99. QS: [merge] Concatenate #2 using a space.
100. QS: [EOQ]
101. QC: {input}
102. Output:
103. QS: <QS-1>
104. QS: <QS-2>
105. .....
106. QS: <QS-N>
107. '''
108. split\_template = '''
109. Q: What are the words in "Alan Mathison Turing"?
110. Answer: ["Alan", "Mathison", "Turing"]
111. Q: What are the words in "John von Neumann"?
112. Answer: ["John", "von", "Neumann"]
113. Q: What are the letters in "Alan"?
114. Answer: ["A", "l", "a", "n"]
115. Q: What are the letters and their positions in "Mathison"?
116. Answer: "[(M, 1), (a, 2), (t, 3), (h, 4), (i, 5), (s, 6), (o, 7), (n, 8)]"
117. Q: What are the words and their positions in "Grace Brewster Murray Hopper"?
118. Answer: "[(Grace, 1), (Brewster, 2), (Murray, 3), (Hopper, 4)]"
119. Q: {input}
120. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
121. Answer: <your answer here>
122. '''
123. merge\_template = '''
124. Q: Concatenate ["A", "l", "a", "n"].
125. Answer: "Alan"
126. Q: Concatenate ["b", "x", "o"] using a space.
127. Answer: "b x o"
128. Q: Concatenate ["a", "a", "g"] using a comma.
129. Answer: "a,a,g"
130. Q: Concatenate ["Alan", "Mathison", "Turing"] using a space.
131. Answer: "Alan Mathison Turing"
132. Q: Concatenate ["Allen", "Institute"].
133. Answer: "AllenInstitute"
134. Q: {input}
135. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
136. Answer: <your answer here>
137. '''
138. str\_position\_template = '''
139. QC: What is the letter at position 1 of the word "Augusta"?
140. QS: (select) [split] What are the letters and their positions in "Augusta"?
141. QS: (select) [arr\_position] What is at position 1 in #1?
142. QS: [EOQ]
143. QC: What is the last letter of the word "Mathison"?
144. QS: (select) [split] What are the letters and their positions in "Mathison"?
145. QS: (select) [arr\_position] What is the last letter in #1?
146. QS: [EOQ]
147. QC: What is the word at the position 4 in "Colorless green ideas sleep furiously"?
148. QS: (select) [split] What are the words and their positions in "Colorless green ideas sleep furiously"?
149. QS: (select) [arr\_position] What is at the position 4 in #1?
150. QS: [EOQ]
151. QC: {input}
152. Output:
153. QS: <QS-1>
154. QS: <QS-2>
155. .....
156. QS: <QS-N>
157. '''
158. arr\_position\_template = '''
159. Q: What is at position 4 in "[("Colorless", 1), ("green", 2), ("ideas", 3), ("sleep", 4), ("furiously", 5)]"?
160. Answer: "sleep"
161. Q: What is at position 1 in "[(M, 1), (a, 2), (t, 3), (h, 4), (i, 5), (s, 6), (o, 7), (n, 8)]"?
162. Answer: "M"
163. Q: What is at the last position in "[(A, 1), (u, 2), (g, 3), (u, 4), (s, 5), (t, 6), (a, 7)]"?
164. Answer: "a"
165. Q: What is at position 1 in "[(Herbert, 1), (Alexander, 2), (Simon, 3)]"?
166. Answer: "Herbert"
167. Q: What is at last position in "[(Allen, 1), (Institute, 2), (for, 3), (Artificial, 4), (Intelligence, 5)]"?
168. Answer: "Intelligence"
169. Q: What is at position 4 in "[(A, 1), (l, 2), (e, 3), (x, 4), (a, 5), (n, 6), (d, 7), (e, 8), (r, 9)]"?
170. Answer: "x"
171. Q: {input}
172. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
173. Answer: <your answer here>
174. '''
175. # prompts/commaQA/template.py
176. decomp\_qa\_template = '''Generate a Question Set (QS) based on the given Question Context (QC) WITHOUT PROVIDING Answer (A). Each QS step should logically follow from the previous one, leading to the final answer, but do not include the answer. Use consistent format and tags as shown below.
177. What awards have movies produced by people born in 1910 won?
178. QS: (select) [simp\_qa] Who were born in the year 1910?
179. QS: (project\_values\_flat\_unique) [pos\_qa] For which movies was #1 the producer?
180. QS: (project\_values\_flat\_unique) [aw\_qa] Which awards were given to #2?
181. QS: [EOQ]
182. QC: What movies have people from the country Stridery acted in?
183. QS: (select) [simp\_qa] Who is from the country Stridery?
184. QS: (project\_values\_flat\_unique) [pos\_qa] Which movies has #1 been an actor in?
185. QS: [EOQ]
186. QC: What awards have the actors of the Erowid winning movies received?
187. QS: (select) [aw\_qa] Which movies were given the Erowid award?
188. QS: (project\_values\_flat\_unique) [pos\_qa] Who are the actors in the movie #1?
189. QS: (project\_values\_flat\_unique) [aw\_qa] Which awards were given to #2?
190. QS: [EOQ]
191. QC: What awards did the movies directed by the Modiparity winners receive?
192. QS: (select) [aw\_qa] Who has been awarded the Modiparity award?
193. QS: (project\_values\_flat\_unique) [pos\_qa] Which movies has #1 directed?
194. QS: (project\_values\_flat\_unique) [aw\_qa] Which awards were given to #2?
195. QS: [EOQ]
196. QC: What awards have movies written by people born in 1935 won?
197. QS: (select) [simp\_qa] Who were born in the year 1935?
198. QS: (project\_values\_flat\_unique) [pos\_qa] What movies has #1 written?
199. QS: (project\_values\_flat\_unique) [aw\_qa] Which awards were given to #2?
200. QS: [EOQ]
201. QC: What movies have the directors from Legault directed?
202. QS: (select) [simp\_qa] Who is from the country Legault?
203. QS: (project\_values\_flat\_unique) [pos\_qa] What movies has #1 been the director of?
204. QS: [EOQ]
205. QC: {input}
206. Output:
207. QS: <QS-1>
208. QS: <QS-2>
209. .....
210. QS: <QS-N>
211. '''
212. pos\_qa\_template = '''
213. movie: Premercy ; directed by: Muntaril. movie: Skirtsicine ; director: Teeplemole. movie: Featsaw ; directed by: Monsterscar. movie: Zalate ; director: Monsterscar. movie: Zalate ; awarded: Hallowcock. movie: Featsaw ; awarded: Zorgion. movie: Premercy ; award: Chowwurst. movie: Skirtsicine ; award: Hallowcock. award: Goatfly ; winner: Teeplemole. person: Monsterscar ; award: Glodome. person: Muntaril ; award: Goatfly. movie: Featsaw ; release year: 1973. movie: Zalate ; release year: 1964. movie: Skirtsicine ; release year: 1973. movie: Premercy ; year: 1961. Teeplemole was an actor in the movie Skirtsicine. Muntaril was an actor in the movie Skirtsicine. Monsterscar was an actor in the movie Premercy. Muntaril was an actor in the movie Featsaw. Teeplemole was an actor in the movie Zalate. Muntaril was born in the year 1910. Teeplemole was born in 1910. Monsterscar was born in 1942. Teeplemole is from the country of Piperfish. Monsterscar is from the country of Piperfish. Muntaril is from the country of Clony. Muntaril produced the movie Skirtsicine with others. Monsterscar was one of the producers of the movie Featsaw. Monsterscar produced the movie Premercy with others. Monsterscar produced the movie Zalate with others. Teeplemole was one of the producers of the movie Featsaw. Teeplemole produced the movie Zalate with others. Muntaril produced the movie Premercy with others. Monsterscar wrote for the movie Premercy. Muntaril was one of the writers for the movie Zalate. Muntaril wrote for the movie Featsaw. Teeplemole wrote for the movie Featsaw. Monsterscar was one of the writers for the movie Zalate. Teeplemole was one of the writers for the movie Skirtsicine.
214. Q: For which movies was Teeplemole the producer?
215. Answer: ["Featsaw", "Zalate"]
216. Q: For which movies was Muntaril the producer?
217. Answer: ["Premercy]
218. movie: Nilitude ; director: Monsterscar. movie: Dewbar ; directed by: Metatoun. movie: Warpstone ; directed by: Gastrat. movie: Partnershipmaker ; director: Metatoun. movie: Dewbar ; award: Tachychronograph. movie: Partnershipmaker ; awarded: Tachychronograph. movie: Nilitude ; award: Paleodactyl. movie: Warpstone ; award: Sabonade. person: Gastrat ; award: Trifogation. award: Polyquadrase ; winner: Monsterscar. award: Trifogation ; winner: Metatoun. movie: Warpstone ; release year: 1956. movie: Dewbar ; release year: 1984. movie: Nilitude ; year: 1984. movie: Partnershipmaker ; year: 1962. Gastrat was an actor in the movie Partnershipmaker. Metatoun was an actor in the movie Partnershipmaker. Metatoun was an actor in the movie Nilitude. Gastrat acted in the movie Nilitude. Monsterscar was an actor in the movie Dewbar. Gastrat acted in the movie Warpstone. Metatoun acted in the movie Warpstone. Metatoun was born in 1939. Gastrat was born in the year 1933. Monsterscar was born in 1933. Metatoun grew up in the nation of Moulole. Gastrat is from the country of Stridery. Monsterscar grew up in the nation of Moulole. Monsterscar produced the movie Nilitude with others. Monsterscar was one of the producers of the movie Warpstone. Metatoun was one of the producers of the movie Warpstone. Gastrat was one of the producers of the movie Nilitude. Metatoun produced the movie Partnershipmaker with others. Metatoun produced the movie Dewbar with others. Monsterscar was one of the producers of the movie Partnershipmaker. Gastrat produced the movie Dewbar with others. Metatoun wrote for the movie Partnershipmaker. Gastrat wrote for the movie Warpstone. Gastrat was one of the writers for the movie Dewbar. Monsterscar was one of the writers for the movie Nilitude. Metatoun wrote for the movie Warpstone.
219. Q: Which movies has Gastrat been an actor in?
220. Answer: ["Partnershipmaker", "Nilitude", "Warpstone"]
221. Q: Which movies has Monsterscar been an actor in?
222. Answer: ["Dewbar"]
223. movie: Pastillobox ; directed by: Firmline. movie: Clenestration ; directed by: Carblock. movie: Pestok ; directed by: Bioperatology. movie: Vitrilateral ; director: Bioperatology. movie: Vitrilateral ; award: Antipositive. movie: Clenestration ; awarded: Handt. movie: Pastillobox ; awarded: Handt. movie: Pestok ; awarded: Gutney. movie: Pestok ; writer: Firmline. movie: Clenestration ; written by: Carblock. movie: Pastillobox ; written by: Bioperatology. movie: Pestok ; writer: Bioperatology. movie: Clenestration ; written by: Firmline. movie: Vitrilateral ; writer: Bioperatology. movie: Pastillobox ; writer: Carblock. movie: Vitrilateral ; written by: Carblock. movie: Pestok ; release year: 1986. movie: Clenestration ; year: 1986. movie: Vitrilateral ; year: 1999. movie: Pastillobox ; release year: 1984. Carblock was an actor in the movie Pastillobox. Firmline was an actor in the movie Vitrilateral. Bioperatology was an actor in the movie Clenestration. Firmline acted in the movie Pastillobox. Carblock was an actor in the movie Clenestration. Bioperatology was an actor in the movie Pestok. Firmline was born in the year 1904. Bioperatology was born in the year 1935. Carblock was born in 1935. Carblock grew up in the nation of Knoppock. Firmline grew up in the nation of Tatkin. Bioperatology grew up in the nation of Tatkin. Bioperatology won the Modiparity award. Halfbill was awarded to Firmline. Halfbill was awarded to Carblock. Bioperatology was one of the producers of the movie Pestok. Bioperatology produced the movie Vitrilateral with others. Firmline produced the movie Pastillobox with others. Firmline produced the movie Clenestration with others. Carblock was one of the producers of the movie Pastillobox. Carblock produced the movie Vitrilateral with others. Carblock produced the movie Clenestration with others. Firmline was one of the producers of the movie Pestok.
224. Q: Which movies has Bioperatology directed?
225. Answer: ["Pestok", "Vitrilateral"]
226. Q: Which movies has Carblock directed?
227. Answer: ["Clenestration"]
228. movie: Nohit ; director: Mimicocycle. movie: Noenometer ; director: Mimicocycle. movie: Tayenne ; directed by: Zayage. movie: Pneumodendron ; director: Sclerocybin. movie: Tayenne ; awarded: Goosehead. movie: Nohit ; awarded: Handt. movie: Pneumodendron ; award: Handt. movie: Noenometer ; awarded: Brownbeard. movie: Nohit ; writer: Mimicocycle. movie: Noenometer ; written by: Sclerocybin. movie: Tayenne ; writer: Sclerocybin. movie: Pneumodendron ; written by: Zayage. movie: Tayenne ; writer: Zayage. movie: Pneumodendron ; written by: Mimicocycle. movie: Noenometer ; release year: 1991. movie: Tayenne ; year: 2013. movie: Nohit ; year: 2005. movie: Pneumodendron ; year: 2005. Mimicocycle was an actor in the movie Tayenne. Zayage acted in the movie Pneumodendron. Zayage was an actor in the movie Nohit. Sclerocybin was an actor in the movie Nohit. Sclerocybin was an actor in the movie Tayenne. Mimicocycle was an actor in the movie Noenometer. Zayage was born in 1935. Sclerocybin was born in 1935. Mimicocycle was born in 1930. Mimicocycle is from the country of Calderita. Sclerocybin grew up in the nation of Calderita. Zayage is from the country of Obility. Quinion was awarded to Zayage. Fannyxist was awarded to Sclerocybin. Fannyxist was awarded to Mimicocycle. Mimicocycle produced the movie Nohit with others. Zayage was one of the producers of the movie Nohit. Sclerocybin was one of the producers of the movie Tayenne. Sclerocybin produced the movie Pneumodendron with others. Zayage produced the movie Pneumodendron with others. Mimicocycle was one of the producers of the movie Tayenne. Sclerocybin was one of the producers of the movie Noenometer. Zayage produced the movie Noenometer with others
229. Q: What movies has Sclerocybin written?
230. Answer: ["Noenometer", "Tayenne"]
231. Q: What movies has Zayage written?
232. Answer: ["Pneumodendron", "Tayenne"]
233. Input:
234. {passage}
235. Q: {input}
236. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
237. Answer: <your answer here>
238. '''
239. aw\_qa\_template = '''
240. movie: Premercy ; directed by: Muntaril. movie: Skirtsicine ; director: Teeplemole. movie: Featsaw ; directed by: Monsterscar. movie: Zalate ; director: Monsterscar. movie: Zalate ; awarded: Hallowcock. movie: Featsaw ; awarded: Zorgion. movie: Premercy ; award: Chowwurst. movie: Skirtsicine ; award: Hallowcock. award: Goatfly ; winner: Teeplemole. person: Monsterscar ; award: Glodome. person: Muntaril ; award: Goatfly. movie: Featsaw ; release year: 1973. movie: Zalate ; release year: 1964. movie: Skirtsicine ; release year: 1973. movie: Premercy ; year: 1961. Teeplemole was an actor in the movie Skirtsicine. Muntaril was an actor in the movie Skirtsicine. Monsterscar was an actor in the movie Premercy. Muntaril was an actor in the movie Featsaw. Teeplemole was an actor in the movie Zalate. Muntaril was born in the year 1910. Teeplemole was born in 1910. Monsterscar was born in 1942. Teeplemole is from the country of Piperfish. Monsterscar is from the country of Piperfish. Muntaril is from the country of Clony. Muntaril produced the movie Skirtsicine with others. Monsterscar was one of the producers of the movie Featsaw. Monsterscar produced the movie Premercy with others. Monsterscar produced the movie Zalate with others. Teeplemole was one of the producers of the movie Featsaw. Teeplemole produced the movie Zalate with others. Muntaril produced the movie Premercy with others. Monsterscar wrote for the movie Premercy. Muntaril was one of the writers for the movie Zalate. Muntaril wrote for the movie Featsaw. Teeplemole wrote for the movie Featsaw. Monsterscar was one of the writers for the movie Zalate. Teeplemole was one of the writers for the movie Skirtsicine.
241. Q: Which awards were given to Zalate?
242. Answer: ["Hallowcock"]
243. Q: Which awards were given to Premercy?
244. Answer: ["Chowwurst"]
245. movie: Misgendery ; directed by: Wetherality. movie: Dewbar ; director: Gigabut. movie: Caudacite ; director: Lougerière. movie: Tayenne ; directed by: Lougerière. movie: Misgendery ; awarded: Microsouenesis. movie: Dewbar ; awarded: Erowid. movie: Tayenne ; awarded: Cockspit. movie: Caudacite ; award: Erowid. award: Aniconder ; winner: Wetherality. award: Aniconder ; winner: Lougerière. person: Gigabut ; award: Trifogation. movie: Dewbar ; release year: 1991. movie: Tayenne ; year: 2013. movie: Caudacite ; release year: 2008. movie: Misgendery ; year: 1991. Wetherality was an actor in the movie Dewbar. Gigabut was an actor in the movie Tayenne. Lougerière was an actor in the movie Tayenne. Lougerière acted in the movie Caudacite. Lougerière acted in the movie Misgendery. Gigabut was an actor in the movie Caudacite. Wetherality was an actor in the movie Misgendery. Wetherality was born in the year 1917. Lougerière was born in 1926. Gigabut was born in the year 1917. Gigabut grew up in the nation of Triclops. Lougerière is from the country of Tatkin. Wetherality grew up in the nation of Tatkin. Lougerière produced the movie Dewbar with others. Gigabut produced the movie Tayenne with others. Gigabut produced the movie Dewbar with others. Lougerière was one of the producers of the movie Misgendery. Wetherality was one of the producers of the movie Caudacite. Gigabut was one of the producers of the movie Caudacite. Wetherality produced the movie Misgendery with others. Wetherality produced the movie Tayenne with others. Wetherality wrote for the movie Tayenne. Gigabut wrote for the movie Misgendery. Lougerière was one of the writers for the movie Caudacite. Wetherality wrote for the movie Misgendery. Gigabut wrote for the movie Tayenne. Gigabut wrote for the movie Dewbar. Lougerière wrote for the movie Dewbar. Wetherality wrote for the movie Caudacite.
246. Q: Which movies were given the Erowid award?
247. Answer: ["Dewbar", "Caudacite"]
248. Q: Which awards were given to Wetherality?
249. Answer: ["Aniconder"]
250. movie: Pastillobox ; directed by: Firmline. movie: Clenestration ; directed by: Carblock. movie: Pestok ; directed by: Bioperatology. movie: Vitrilateral ; director: Bioperatology. movie: Vitrilateral ; award: Antipositive. movie: Clenestration ; awarded: Handt. movie: Pastillobox ; awarded: Handt. movie: Pestok ; awarded: Gutney. movie: Pestok ; writer: Firmline. movie: Clenestration ; written by: Carblock. movie: Pastillobox ; written by: Bioperatology. movie: Pestok ; writer: Bioperatology. movie: Clenestration ; written by: Firmline. movie: Vitrilateral ; writer: Bioperatology. movie: Pastillobox ; writer: Carblock. movie: Vitrilateral ; written by: Carblock. movie: Pestok ; release year: 1986. movie: Clenestration ; year: 1986. movie: Vitrilateral ; year: 1999. movie: Pastillobox ; release year: 1984. Carblock was an actor in the movie Pastillobox. Firmline was an actor in the movie Vitrilateral. Bioperatology was an actor in the movie Clenestration. Firmline acted in the movie Pastillobox. Carblock was an actor in the movie Clenestration. Bioperatology was an actor in the movie Pestok. Firmline was born in the year 1904. Bioperatology was born in the year 1935. Carblock was born in 1935. Carblock grew up in the nation of Knoppock. Firmline grew up in the nation of Tatkin. Bioperatology grew up in the nation of Tatkin. Bioperatology won the Modiparity award. Halfbill was awarded to Firmline. Halfbill was awarded to Carblock. Bioperatology was one of the producers of the movie Pestok. Bioperatology produced the movie Vitrilateral with others. Firmline produced the movie Pastillobox with others. Firmline produced the movie Clenestration with others. Carblock was one of the producers of the movie Pastillobox. Carblock produced the movie Vitrilateral with others. Carblock produced the movie Clenestration with others. Firmline was one of the producers of the movie Pestok.
251. Q: Who has been awarded the Modiparity award?
252. Answer: ["Bioperatology"]
253. Q: Which awards were given to Vitrilateral?
254. Answer: ["Antipositive"]
255. movie: Nohit ; directo: Mimicocycle. movie: Noenometer ; director: Mimicocycle. movie: Tayenne ; directed by: Zayage. movie: Pneumodendron ; director: Sclerocybin. movie: Tayenne ; awarded: Goosehead. movie: Nohit ; awarded: Handt. movie: Pneumodendron ; award: Handt. movie: Noenometer ; awarded: Brownbeard. movie: Nohit ; writer: Mimicocycle. movie: Noenometer ; written by: Sclerocybin. movie: Tayenne ; writer: Sclerocybin. movie: Pneumodendron ; written by: Zayage. movie: Tayenne ; writer: Zayage. movie: Pneumodendron ; written by: Mimicocycle. movie: Noenometer ; release year: 1991. movie: Tayenne ; year: 2013. movie: Nohit ; year: 2005. movie: Pneumodendron ; year: 2005. Mimicocycle was an actor in the movie Tayenne. Zayage acted in the movie Pneumodendron. Zayage was an actor in the movie Nohit. Sclerocybin was an actor in the movie Nohit. Sclerocybin was an actor in the movie Tayenne. Mimicocycle was an actor in the movie Noenometer. Zayage was born in 1935. Sclerocybin was born in 1935. Mimicocycle was born in 1930. Mimicocycle is from the country of Calderita. Sclerocybin grew up in the nation of Calderita. Zayage is from the country of Obility. Quinion was awarded to Zayage. Fannyxist was awarded to Sclerocybin. Fannyxist was awarded to Mimicocycle. Mimicocycle produced the movie Nohit with others. Zayage was one of the producers of the movie Nohit. Sclerocybin was one of the producers of the movie Tayenne. Sclerocybin produced the movie Pneumodendron with others. Zayage produced the movie Pneumodendron with others. Mimicocycle was one of the producers of the movie Tayenne. Sclerocybin was one of the producers of the movie Noenometer. Zayage produced the movie Noenometer with others
256. Q: Which awards were given to Tayenne?
257. Answer: ["Goosehead"]
258. Q: Which awards were given to Noenometer?
259. Answer: ["Brownbeard"]
260. Input:
261. {passage}
262. Q: {input}
263. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
264. Answer: <your answer here>
265. '''
266. simp\_qa\_template = '''
267. movie: Premercy ; directed by: Muntaril. movie: Skirtsicine ; director: Teeplemole. movie: Featsaw ; directed by: Monsterscar. movie: Zalate ; director: Monsterscar. movie: Zalate ; awarded: Hallowcock. movie: Featsaw ; awarded: Zorgion. movie: Premercy ; award: Chowwurst. movie: Skirtsicine ; award: Hallowcock. award: Goatfly ; winner: Teeplemole. person: Monsterscar ; award: Glodome. person: Muntaril ; award: Goatfly. movie: Featsaw ; release year: 1973. movie: Zalate ; release year: 1964. movie: Skirtsicine ; release year: 1973. movie: Premercy ; year: 1961. Teeplemole was an actor in the movie Skirtsicine. Muntaril was an actor in the movie Skirtsicine. Monsterscar was an actor in the movie Premercy. Muntaril was an actor in the movie Featsaw. Teeplemole was an actor in the movie Zalate. Muntaril was born in the year 1910. Teeplemole was born in 1910. Monsterscar was born in 1942. Teeplemole is from the country of Piperfish. Monsterscar is from the country of Piperfish. Muntaril is from the country of Clony. Muntaril produced the movie Skirtsicine with others. Monsterscar was one of the producers of the movie Featsaw. Monsterscar produced the movie Premercy with others. Monsterscar produced the movie Zalate with others. Teeplemole was one of the producers of the movie Featsaw. Teeplemole produced the movie Zalate with others. Muntaril produced the movie Premercy with others. Monsterscar wrote for the movie Premercy. Muntaril was one of the writers for the movie Zalate. Muntaril wrote for the movie Featsaw. Teeplemole wrote for the movie Featsaw. Monsterscar was one of the writers for the movie Zalate. Teeplemole was one of the writers for the movie Skirtsicine.
268. Q: Who were born in the year 1910?
269. Answer: ["Teeplemole", "Muntaril"]
270. Q: Who is from the country Piperfish?
271. Answer: ["Teeplemole", "Monsterscar"]
272. movie: Nilitude ; director: Monsterscar. movie: Dewbar ; directed by: Metatoun. movie: Warpstone ; directed by: Gastrat. movie: Partnershipmaker ; director: Metatoun. movie: Dewbar ; award: Tachychronograph. movie: Partnershipmaker ; awarded: Tachychronograph. movie: Nilitude ; award: Paleodactyl. movie: Warpstone ; award: Sabonade. person: Gastrat ; award: Trifogation. award: Polyquadrase ; winner: Monsterscar. award: Trifogation ; winner: Metatoun. movie: Warpstone ; release year: 1956. movie: Dewbar ; release year: 1984. movie: Nilitude ; year: 1984. movie: Partnershipmaker ; year: 1962. Gastrat was an actor in the movie Partnershipmaker. Metatoun was an actor in the movie Partnershipmaker. Metatoun was an actor in the movie Nilitude. Gastrat acted in the movie Nilitude. Monsterscar was an actor in the movie Dewbar. Gastrat acted in the movie Warpstone. Metatoun acted in the movie Warpstone. Metatoun was born in 1939. Gastrat was born in the year 1933. Monsterscar was born in 1933. Metatoun grew up in the nation of Moulole. Gastrat is from the country of Stridery. Monsterscar grew up in the nation of Moulole. Monsterscar produced the movie Nilitude with others. Monsterscar was one of the producers of the movie Warpstone. Metatoun was one of the producers of the movie Warpstone. Gastrat was one of the producers of the movie Nilitude. Metatoun produced the movie Partnershipmaker with others. Metatoun produced the movie Dewbar with others. Monsterscar was one of the producers of the movie Partnershipmaker. Gastrat produced the movie Dewbar with others. Metatoun wrote for the movie Partnershipmaker. Gastrat wrote for the movie Warpstone. Gastrat was one of the writers for the movie Dewbar. Monsterscar was one of the writers for the movie Nilitude. Metatoun wrote for the movie Warpstone.
273. Q: Who is from the country Stridery?
274. Answer: ["Gastrat"]
275. Q: Who were born in the year 1939?
276. Answer: ["Metatoun"]
277. movie: Coacheship ; director: Metatoun. movie: Assamplifier ; director: Kapod. movie: Misapportionment ; director: Sapien. movie: Quinsid ; director: Kapod. movie: Assamplifier ; award: Zorgion. movie: Quinsid ; awarded: Airpipe. movie: Coacheship ; award: Electrodesal. movie: Misapportionment ; award: Airpipe. movie: Coacheship ; written by: Metatoun. movie: Misapportionment ; written by: Kapod. movie: Coacheship ; written by: Kapod. movie: Quinsid ; writer: Sapien. movie: Misapportionment ; written by: Metatoun. movie: Assamplifier ; written by: Kapod. movie: Assamplifier ; written by: Sapien. movie: Assamplifier ; release year: 2000. movie: Coacheship ; year: 2001. movie: Quinsid ; year: 2005. movie: Misapportionment ; year: 2005. Sapien was an actor in the movie Misapportionment. Sapien acted in the movie Assamplifier. Kapod acted in the movie Quinsid. Sapien acted in the movie Coacheship. Metatoun was an actor in the movie Quinsid. Kapod acted in the movie Misapportionment. Metatoun acted in the movie Coacheship. Kapod acted in the movie Assamplifier. Sapien was born in the year 1910. Metatoun was born in 1928. Kapod was born in the year 1910. Metatoun is from the country of Legault. Kapod grew up in the nation of Tatkin. Sapien is from the country of Legault. Malwarp was awarded to Sapien. Metatoun won the Monkeynote award. Kapod won the Monkeynote award. Kapod was one of the producers of the movie Quinsid. Metatoun was one of the producers of the movie Misapportionment. Metatoun was one of the producers of the movie Quinsid. Sapien was one of the producers of the movie Assamplifier. Sapien produced the movie Coacheship with others. Metatoun was one of the producers of the movie Assamplifier. Kapod was one of the producers of the movie Misapportionment. Kapod was one of the producers of the movie Coacheship.
278. Q: Who is from the country Legault?
279. Answer: ["Metatoun", "Sapien"]
280. Q: Who were born in the year 1910?
281. Answer: ["Kapod", "Sapien"]
282. movie: Nohit ; director: Mimicocycle. movie: Noenometer ; director: Mimicocycle. movie: Tayenne ; directed by: Zayage. movie: Pneumodendron ; director: Sclerocybin. movie: Tayenne ; awarded: Goosehead. movie: Nohit ; awarded: Handt. movie: Pneumodendron ; award: Handt. movie: Noenometer ; awarded: Brownbeard. movie: Nohit ; writer: Mimicocycle. movie: Noenometer ; written by: Sclerocybin. movie: Tayenne ; writer: Sclerocybin. movie: Pneumodendron ; written by: Zayage. movie: Tayenne ; writer: Zayage. movie: Pneumodendron ; written by: Mimicocycle. movie: Noenometer ; release year: 1991. movie: Tayenne ; year: 2013. movie: Nohit ; year: 2005. movie: Pneumodendron ; year: 2005. Mimicocycle was an actor in the movie Tayenne. Zayage acted in the movie Pneumodendron. Zayage was an actor in the movie Nohit. Sclerocybin was an actor in the movie Nohit. Sclerocybin was an actor in the movie Tayenne. Mimicocycle was an actor in the movie Noenometer. Zayage was born in 1935. Sclerocybin was born in 1935. Mimicocycle was born in 1930. Mimicocycle is from the country of Calderita. Sclerocybin grew up in the nation of Calderita. Zayage is from the country of Obility. Quinion was awarded to Zayage. Fannyxist was awarded to Sclerocybin. Fannyxist was awarded to Mimicocycle. Mimicocycle produced the movie Nohit with others. Zayage was one of the producers of the movie Nohit. Sclerocybin was one of the producers of the movie Tayenne. Sclerocybin produced the movie Pneumodendron with others. Zayage produced the movie Pneumodendron with others. Mimicocycle was one of the producers of the movie Tayenne. Sclerocybin was one of the producers of the movie Noenometer. Zayage produced the movie Noenometer with others
283. Q: Who were born in the year 1935?
284. Answer: ["Sclerocybin", "Zayage"]
285. Q: Who is from the country Obility?
286. Answer: ["Zayage"]
287. Input:
288. {passage}
289. Q: {input}
290. Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
291. Answer: <your answer here>
292. '''