```python
# utils/gemini_client.py

import os
import google.generativeai as genai
from dotenv import load_dotenv

load_dotenv()

API_KEY = os.getenv("GEMINI_API_KEY")

if not API_KEY:
    raise ValueError("GEMINI_API_KEY not found in environment variables.")

genai.configure(api_key=API_KEY)

model = genai.GenerativeModel("gemini-2.0-flash")

def generate_answer(prompt: str) -> str:
    try:
        response = model.generate_content(
            prompt,
            generation_config={"temperature": 0.7}
        )
        return response.text.strip()
    except Exception as e:
        return f"[ERROR] {str(e)}"
```

```python
# utils/io_utils.py

import json

def load_json(path: str):
    with open(path, "r", encoding="utf-8") as f:
        return json.load(f)

def save_json(data, path: str):
    with open(path, "w", encoding="utf-8") as f:
        json.dump(data, f, indent=2, ensure_ascii=False)
```

```python
# main.py

from utils.io_utils import load_json, save_json
from utils.gemini_client import generate_answer
import re
import os
```

```
import argparse
import pandas as pd
import time

def parse_args():
    parser = argparse.ArgumentParser(description="Run prompting with Gemini API.")
    parser.add_argument(
        "--output",
        type=str,
        default="generate_res/dummy.json",
        help="Output file path (default: generate_res/dummy.json)"
    )
    parser.add_argument(
        "--task",
        type=str,
        default="commaqa",
        help="Dataset type ? (default: commaqa)"
    )
    return parser.parse_args()

def get_controller(task):
    if task == "reverse":
        from reverse_controller import ReverseController
        return ReverseController()
    elif task == "lettercat":
        from lettercat_controller import LatterCatController
        return LatterCatController()
    elif task == "commaqa":
        from commaqa_controller import CommAQAController
        return CommAQAController()
    else:
        raise ValueError(f"Unsupported task: {task}")

def main():
    args = parse_args()
    print("Arguments parsed:")
    for arg, value in vars(args).items():
        print(f"{arg}: {value}")
    print("\n")

    controller = get_controller(args.task)

    # get decomposition chains
    controller.solve()

if __name__ == "__main__":
    main()
```

```python
# reverse_controller.py

from utils.io_utils import load_json, save_json
import prompts.reverse.template as reverse_template
from utils.gemini_client import generate_answer
import re
import os
import time

class ReverseController:
    def __init__(self):
        self.dataset = load_json('dataset/reverse.json')

    def print_questions(self):
        """Prints all questions from the dataset."""
        for entry in self.dataset:
            print(entry['question'])

    def replace_references(self, s, result_array):
        def replacer(match):
            index = int(match.group(1))-1  # Ambil angka dari #3 misalnya
            if 0 <= index < len(result_array):
                return result_array[index]
            else:
                return f"<Invalid index #{index}>"

        # Ganti semua #angka dengan nilai yang sesuai
        return re.sub(r"#(\d+)", replacer, s)

    def extract_qs(self, text):
        matches = re.findall(r'.* (?:is|reverse) \"(.*)\"\.', text)
        return matches[0] if matches else ""

    def get_generated_answer(self, ans):
        if "Answer:" in ans:
            process_answer = ans.split("Answer: ", 1)[1].strip()
        else:
            process_answer = ans.strip()
        return process_answer

    def chain_processing(self, qs_lines):
        result_answers = []

        for index, qs in enumerate(qs_lines):
            match = re.match(r"\[(\w+)]", qs)
            match = match.group(1)
```

```
                process_answer = ""

                if match == "extract":
                    process_answer = self.extract_qs(qs)

                elif match == "remove_numbers":
                    formatted_qs = self.replace_references(qs, result_answers)
                    rm_num_prompt = reverse_template.remove_numbers_template
                                    .replace('{input}', formatted_qs)
                    generated_ans = generate_answer(rm_num_prompt)
                    print(f"Generated answer for remove_numbers: {generated_ans}")
                    process_answer = self.get_generated_answer(generated_ans)

                elif match == "reverse":
                    formatted_qs = self.replace_references(qs, result_answers)
                    cot_prompt = reverse_template.cot_template.replace('{input}',
                                 formatted_qs)
                    print(f"COT prompt for reverse: {cot_prompt}")
                    generated_ans = generate_answer(cot_prompt)
                    print(f"Generated answer for reverse: {generated_ans}")
                    process_answer = self.get_generated_answer(generated_ans)

                elif match == "join":
                    formatted_qs = self.replace_references(qs, result_answers)
                    join_prompt = reverse_template.join_template.replace('{input}',
                                  formatted_qs)
                    generated_ans = generate_answer(join_prompt)
                    process_answer = self.get_generated_answer(generated_ans)

                elif match == "EOQ":
                    break

                print(f"Processing step {index+1}: {match} -> {process_answer}")
                result_answers.append(process_answer)

            return result_answers

        def generate_chain(self, question):
            template = reverse_template.decomp_chain
            prompt = template.replace('{input}', question)
            decomp_chain = generate_answer(prompt)

            qs_lines = [line.replace("QS: ", "").strip() for line in
                        decomp_chain.splitlines() if line.startswith("QS:")]
            print(f"Decomposition chain generated: {qs_lines}")
            return qs_lines

        def solve(self):
```

```
            output_log = []

            for entry in self.dataset:

                qs_lines = self.generate_chain(entry['question'])
                output = self.chain_processing(qs_lines)

                output_log.append({
                    'question': entry['question'],
                    'qs_lines': qs_lines,
                    'result': output,
                    "is_correct": entry['answer']["spans"][0] == output[-1] if
                                        output else False
                })

                save_json(output_log, 'generate_res/dummy.json')
                time.sleep(60)  # To avoid hitting rate limits

            return True
```

```
# lettercat_controller.py

from utils.io_utils import load_json, save_json
import prompts.lettercat.template as lettercat_template
from utils.gemini_client import generate_answer
import re
import os
import time
import json

class LatterCatController:
    def __init__(self):
        self.dataset = load_json('dataset/lettercat.json')

    def print_questions(self):
        """Prints all questions from the dataset."""
        for entry in self.dataset:
            print(entry['question'])

    def replace_references(self, s, result_array):
        def replacer(match):
            index = int(match.group(1))-1  # Ambil angka dari #3 misalnya
            if 0 <= index < len(result_array):
                return result_array[index]
            else:
                return f"<Invalid index #{index}>"

        # Ganti semua #angka dengan nilai yang sesuai
```

```
27.             return re.sub(r"#(\d+)", replacer, s)
28.
29.         def get_references(self, qs):
30.             references = re.findall(r'#(\d+)', qs)
31.             return [int(ref) for ref in references]
32.
33.         def get_generated_answer(self, ans):
34.             if "Answer:" in ans:
35.                 process_answer = ans.split("Answer: ", 1)[1].strip()
36.             else:
37.                 process_answer = ans.strip()
38.             return process_answer
39.
40.         def chain_processing(self, qs_lines):
41.             result_answers = []
42.
43.             for index, qs in enumerate(qs_lines):
44.                 match = re.search(r'\[([^\]]+)\]', qs)
45.                 match = match.group(1)
46.                 process_answer = ""
47.
48.                 if match == "split":
49.                     split_prompt = lettercat_template.split_template.replace('{input}',
                        qs)
50.                     generated_ans = generate_answer(split_prompt)
51.                     print(f"Generated answer for split: {generated_ans}")
52.                     process_answer = self.get_generated_answer(generated_ans)
53.
54.                 elif match == "str_position":
55.                     references = self.get_references(qs)
56.                     word_lists = json.loads(result_answers[references[0]-1])
57.                     ans_list = []
58.
59.                     time.sleep(60)
60.
61.                     for word in word_lists:
62.                         formatted_qs = re.sub(r"#\d+", word, qs)
63.                         print("Formatted qs: ")
64.                         print(formatted_qs)
65.                         cleaned_formatted_qs = re.sub(r'\(.*?\)|\[.*?\]', '',
                                                    formatted_qs).strip()
66.                         print("Cleaned formatted qs: ")
67.                         print(cleaned_formatted_qs)
68.
69.                         str_pos_decomp_chain = self.generate_chain(lettercat_template
                                        .str_position_template, cleaned_formatted_qs)
70.                         ans_list.append(self.chain_processing(str_pos_decomp_chain)[-
                                    1])
```

```
71.
72.                     print(f"Split Result: {ans_list}")
73.                     print(ans_list)
74.                     process_answer = str(ans_list)
75.
76.                 elif match == "arr_position":
77.                     formatted_qs = self.replace_references(qs, result_answers)
78.                     arr_pos_prompt = lettercat_template.arr_position_template
                                        .replace('{input}', formatted_qs)
79.                     generated_ans = generate_answer(arr_pos_prompt)
80.                     process_answer = self.get_generated_answer(generated_ans)
81.
82.                 elif match == "merge":
83.                     formatted_qs = self.replace_references(qs, result_answers)
84.                     print("Merge Process")
85.                     print(formatted_qs)
86.                     merge_prompt = lettercat_template.merge_template.replace('{input}',
                                    formatted_qs)
87.                     generated_ans = generate_answer(merge_prompt)
88.                     print(f"Generated answer for merge: {generated_ans}")
89.                     process_answer = self.get_generated_answer(generated_ans)
90.
91.                 elif match == "EOQ":
92.                     break
93.
94.                 # print(f"Processing step {index+1}: {match} -> {process_answer}")
95.                 result_answers.append(process_answer)
96.
97.             return result_answers
98.
99.         def generate_chain(self, template, question):
100.            prompt = template.replace('{input}', question)
101.            decomp_chain = generate_answer(prompt)
102.
103.            qs_lines = [line.replace("QS: ", "").strip() for line in
                           decomp_chain.splitlines() if line.startswith("QS:")]
104.            return qs_lines
105.
106.        def solve(self):
107.
108.            output_log = []
109.            for entry in self.dataset:
110.
111.                print(f"Processing question: {entry['question']}")
112.                template = lettercat_template.decomp_template
113.                qs_lines = self.generate_chain(template, entry['question'])
114.                output = self.chain_processing(qs_lines)
115.
```

```
116.                output_log.append({
117.                    'question': entry['question'],
118.                    'qs_lines': qs_lines,
119.                    'result': output,
120.                    "is_correct": entry['answer']["spans"][0] == output[-1] if
                                                                    output else False
121.                })
122.
123.                save_json(output_log, 'generate_res/dummy.json')
124.                time.sleep(60)  # To avoid hitting rate limits
125.
126.            return True
```

```
1.      # commaqa_controller.py
2.
3.      from utils.io_utils import load_json, save_json
4.      import prompts.commaQA.template as commaQA_template
5.      from utils.gemini_client import generate_answer
6.      import re
7.      import os
8.      import time
9.      import json
10.     import ast
11.
12.     class CommAQAController:
13.         def __init__(self):
14.             self.dataset = load_json('dataset/commaqa.json')
15.
16.         def print_questions(self):
17.             """Prints all questions from the dataset."""
18.             for entry in self.dataset:
19.                 print(entry['question'])
20.
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}>"
28.
29.             # Ganti semua #angka dengan nilai yang sesuai
30.             return re.sub(r"#(\d+)", replacer, s)
31.
32.         def get_references(self, qs):
33.             references = re.findall(r'#(\d+)', qs)
34.             return [int(ref) for ref in references]
35.
```

```
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
42.
43.         def get_Qproject_values_flat_unique(self, result_array):
44.             return
45.
46.         def clean_text(self, text):
47.             return re.sub(r'\(.*?\)|\[.*?\]', '', text).strip()
48.
49.         def get_references_array(self, qs, result_answers):
50.             references = result_answers[self.get_references(qs)[0]-1]
51.             print(f"references: {references}")
52.             array_ver = json.loads(references)
53.             print(array_ver)
54.             return array_ver
55.
56.         def get_score(self, output, entry_answer):
57.             ans = ast.literal_eval(output[-1])
58.             ans_expected = entry_answer["spans"]
59.             total_number = len(ans_expected)
60.
61.             temp_score = 0
62.             for ans_exp in ans_expected:
63.                 if ans_exp in ans:
64.                     temp_score += 1
65.
66.             return temp_score/total_number
67.
68.         def simple_subtask(self, qs, prompt_template, passage=""):
69.             cleaned_text = self.clean_text(qs)
70.             prompt = prompt_template.replace('{input}', cleaned_text)
                                        .replace('{passage}', passage)
71.             generated_ans = generate_answer(prompt)
72.             return self.get_generated_answer(generated_ans)
73.
74.         def project_values_flat_unique_subtask(self, qs, result_answers,
                                                   prompt_template, passage=""):
75.             references = self.get_references_array(qs, result_answers)
76.             ans_list = []
77.             print(f"QA Prompt References: {references}")
78.             for ref in references:
79.                 # format and generate prompt template
80.                 formatted_qs = re.sub(r"#\d+", ref, qs)
81.                 cleaned_text = self.clean_text(formatted_qs)
```

```
82.                 prompt = prompt_template.replace('{input}', cleaned_text)
                                            .replace('{passage}', passage)
83.                 # generate answer
84.                 generated_ans = generate_answer(prompt)
85.                 ans_list.append(json.loads(self.get_generated_answer(generated_ans)))
86.
87.             # convert to SET DataStructure add to process answer
88.             ans_set_list = list(set(item for sublist in ans_list for item in sublist))
89.             return json.dumps(ans_set_list)
90.
91.         def chain_processing(self, qs_lines, passage=""):
92.             result_answers = []
93.
94.             for index, qs in enumerate(qs_lines):
95.                 # Cari yang dalam tanda kurung bulat (command)
96.                 command_match = re.search(r"\[(.*?)\]", qs)
97.                 command = command_match.group(1) if command_match else ''
98.
99.                 # Cari yang dalam tanda kurung kotak (type)
100.                type_match = re.search(r"\((.*?)\)", qs)
101.                query_type = type_match.group(1) if type_match else ''
102.                process_answer = ""
103.
104.                if command == "simp_qa":
105.                    if query_type == "select":
106.                        process_answer = self.simple_subtask(qs, commaQA_template
                                                .simp_qa_template, passage)
107.
108.                    elif query_type == "project_values_flat_unique":
109.                        process_answer = self.project_values_flat_unique_subtask(qs,
                                                result_answers,
                                                commaQA_template.simp_qa_template, passage)
110.
111.                elif command == "pos_qa":
112.                    if query_type == "select":
113.                        process_answer = self.simple_subtask(qs,
                                                commaQA_template.pos_qa_template, passage)
114.
115.                    elif query_type == "project_values_flat_unique":
116.                        process_answer = self.project_values_flat_unique_subtask(qs,
                                                result_answers,
                                                commaQA_template.pos_qa_template, passage)
117.
118.                elif command == "aw_qa":
119.                    if query_type == "select":
120.                        process_answer = self.simple_subtask(qs,
                                               commaQA_template.aw_qa_template, passage)
121.
```

```
                elif query_type == "project_values_flat_unique":
                    process_answer = self.project_values_flat_unique_subtask(qs,
                    result_answers, commaQA_template.aw_qa_template, passage)

            elif command == "EOQ":
                Break

            print(f"Processing step {index+1}: {command} -> {process_answer}")
            result_answers.append(process_answer)

        return result_answers

    def generate_chain(self, question):
        template = commaQA_template.decomp_qa_template
        prompt = template.replace('{input}', question)
        decomp_chain = generate_answer(prompt)

        qs_lines = [line.replace("QS: ", "").strip() for line in
                    decomp_chain.splitlines() if line.startswith("QS:")]
        return qs_lines

    def solve(self):
        output_log = []
        score = 0

        for key, value in self.dataset.items():
          for index, entry in enumerate(value.get('qa_pairs')):
            qs_lines = self.generate_chain(entry['question'])
            print(f"PROCESS START - {key} - {index}")
            print(f"lines: {qs_lines}")
            output = self.chain_processing(qs_lines,
                    passage=value.get('passage'))

            current_score = self.get_score(output, entry['answer'])
            output_log.append({
                'passage': value.get('passage'),
                'question': entry['question'],
                'qs_lines': qs_lines,
                'result': output,
                "score": current_score
            })
            print(f"score : {current_score}\n")
            score += current_score
            save_json(output_log, 'generate_res/dummy.json')

            time.sleep(60)  # To avoid hitting rate limits
        print(f"final score: {score}")
        return True
```

```
# prompts/reverse/template.py

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.

QC: Reverse the sequence "driving license, button, packet, identity card, shoe".
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".
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".
QS: [remove_numbers] Remove the numbers from #1.
QS: [remove_numbers] Remove the numbers from #2.
QS: [reverse] Reverse the sequence #3.
QS: [reverse] Reverse the sequence #4.
QS: [join] #6 #5
QS: [EOQ]

QC: Reverse the sequence "newspaper, glasses, laptop, bottle".
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".
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".
QS: [remove_numbers] Remove the numbers from #1.
QS: [remove_numbers] Remove the numbers from #2.
QS: [reverse] Reverse the sequence #3.
QS: [reverse] Reverse the sequence #4.
QS: [join] #6 #5
QS: [EOQ]

QC: {input}

Output:
QS: <QS-1>
```

```
    QS: <QS-2>
    .....
    QS: <QS-N>
    '''
    join_template = '''
    Q: "bottle, laptop" "glasses, newspaper".
    Answer: bottle, laptop, glasses, newspaper

    Q: "identity card, packet, button" "magazine, notebook, glasses".
    Answer: identity card, packet, button, magazine, notebook, glasses

    Q: "passport, umbrella, radio, mobile phone, photo" "player".
    Answer:  passport, umbrella, radio, mobile phone, photo, player

    Q: "mirror, case" "toothbrush, alarm clock".
    Answer: mirror, case, toothbrush, alarm clock

    Q: "light bulb, clip, umbrella" "driving licence, watch".
    Answer: light bulb, clip, umbrella, driving licence, watch

    Q: {input}

    Output (your answer MUST be in the same format "Answer: <your answer here>", you
    MUST add "Answer: " before your answer):
    Answer: <your answer here>
    '''

    remove_numbers_template = '''
    Q: Remove the numbers from "4. bottle, 3. laptop, 2. glasses, 1. newspaper".
    Answer: bottle, laptop, glasses, newspaper

    Q: Remove the numbers from "1. identity card, 2. packet, 3. button".
    Answer: identity card, packet, button

    Q: Remove the numbers from "1. player, 2. passport, 3. umbrella, 4. radio".
    Answer: player, passport, umbrella, radio

    Q: {input}

    Output (your answer MUST be in the same format "Answer: <your answer here>", you
    MUST add "Answer: " before your answer):
    Answer: <your answer here>
    '''

    cot_template = '''
    QC: Reverse the sequence "newspaper, glasses, laptop, bottle".
```

```
72.     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".
73.     Answer: bottle, laptop, glasses, newspaper
74.
75.     QC: Reverse the sequence "laptop, photo, clip".
76.     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".
77.     Answer: clip, photo, laptop
78.
79.     QC: Reverse the sequence "driving license, button, packet, identity card,
        pineapple".
80.     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".
81.     Answer: pineapple, identity card, packet, button, driving license
82.
83.     Q: {nput}
84.     QS: <your qs answer here>
85.
86.     Output (your answer MUST be in the same format "Answer: <your answer here>", you
        MUST add "Answer: " before your answer):
87.     Answer: <your final answer here>
88.     '''
```

```
1.      # prompts/lettercat/template.py
2.
3.      decomp_template = '''
4.      QC: Take the last letters of the words in "Elon Musk Tesla" and concatenate them.
5.      QS: [split] What are the words in "Elon Musk"?
6.      QS: (project_values) [str_position] What is the last letter in "#1"?
7.      QS: [merge] Concatenate #2 using a space.
8.      QS: [EOQ]
9.
10.     QC: Take the letters at position 1 of the words in "Sundar Pichai Google" and
        concatenate them.
11.     QS: [split] What are the words in "Sundar Pichai Google"?
12.     QS: (project_values) [str_position] What is the letter at position 1 in "#1"?
13.     QS: [merge] Concatenate #2 using a space.
14.     QS: [EOQ]
15.
16.     QC: Take the letters at position 4 of the words in "Oren Etzioni AllenInstitute"
        and concatenate them.
17.     QS: [split] What are the words in "Oren Etzioni AllenInstitute"?
18.     QS: (project_values) [str_position] What is the letter at position 4 in "#1"?
```

```
    QS: [merge] Concatenate #2 using a space.
    QS: [EOQ]

    QC: {input}

    Output:
    QS: <QS-1>
    QS: <QS-2>
    .....
    QS: <QS-N>
    '''

    split_template = '''
    Q: What are the words in "Alan Mathison Turing"?
    Answer: ["Alan", "Mathison", "Turing"]

    Q: What are the words in "John von Neumann"?
    Answer: ["John", "von", "Neumann"]

    Q: What are the letters in "Alan"?
    Answer: ["A", "l", "a", "n"]

    Q: What are the letters and their positions in "Mathison"?
    Answer: "[(M, 1), (a, 2), (t, 3), (h, 4), (i, 5), (s, 6), (o, 7), (n, 8)]"

    Q: What are the words and their positions in "Grace Brewster Murray Hopper"?
    Answer: "[(Grace, 1), (Brewster, 2), (Murray, 3), (Hopper, 4)]"

    Q: {input}

    Output (your answer MUST be in the same format "Answer: <your answer here>", you
    MUST add "Answer: " before your answer):
    Answer: <your answer here>
    '''

    merge_template = '''
    Q: Concatenate ["A", "l", "a", "n"].
    Answer: "Alan"

    Q: Concatenate ["b", "x", "o"] using a space.
    Answer: "b x o"

    Q: Concatenate ["a", "a", "g"] using a comma.
    Answer: "a,a,g"

    Q: Concatenate ["Alan", "Mathison", "Turing"] using a space.
    Answer: "Alan Mathison Turing"

```

```
    Q: Concatenate ["Allen", "Institute"].
    Answer: "AllenInstitute"

    Q: {input}

    Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
    Answer: <your answer here>
    '''

    str_position_template = '''
    QC: What is the letter at position 1 of the word "Augusta"?
    QS: (select) [split] What are the letters and their positions in "Augusta"?
    QS: (select) [arr_position] What is at position 1 in #1?
    QS: [EOQ]

    QC: What is the last letter of the word "Mathison"?
    QS: (select) [split] What are the letters and their positions in "Mathison"?
    QS: (select) [arr_position] What is the last letter in #1?
    QS: [EOQ]

    QC: What is the word at the position 4 in "Colorless green ideas sleep furiously"?
    QS: (select) [split] What are the words and their positions in "Colorless green ideas sleep furiously"?
    QS: (select) [arr_position] What is at the position 4 in #1?
    QS: [EOQ]

    QC: {input}

    Output:
    QS: <QS-1>
    QS: <QS-2>
    .....
    QS: <QS-N>
    '''

    arr_position_template = '''
    Q: What is at position 4 in "[("Colorless", 1), ("green", 2), ("ideas", 3), ("sleep", 4), ("furiously", 5)]"?
    Answer: "sleep"

    Q: What is at position 1 in "[(M, 1), (a, 2), (t, 3), (h, 4), (i, 5), (s, 6), (o, 7), (n, 8)]"?
    Answer: "M"

    Q: What is at the last position in "[(A, 1), (u, 2), (g, 3), (u, 4), (s, 5), (t, 6), (a, 7)]"?
    Answer: "a"
```

```
Q: What is at position 1 in "[(Herbert, 1), (Alexander, 2), (Simon, 3)]"?
Answer: "Herbert"

Q: What is at last position in "[(Allen, 1), (Institute, 2), (for, 3), (Artificial, 4), (Intelligence, 5)]"?
Answer: "Intelligence"

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)]"?
Answer: "x"

Q: {input}

Output (your answer MUST be in the same format "Answer: <your answer here>", you MUST add "Answer: " before your answer):
Answer: <your answer here>
'''
```

```
# prompts/commaQA/template.py

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.

What awards have movies produced by people born in 1910 won?
QS: (select) [simp_qa] Who were born in the year 1910?
QS: (project_values_flat_unique) [pos_qa] For which movies was #1 the producer?
QS: (project_values_flat_unique) [aw_qa] Which awards were given to #2?
QS: [EOQ]

QC: What movies have people from the country Stridery acted in?
QS: (select) [simp_qa] Who is from the country Stridery?
QS: (project_values_flat_unique) [pos_qa] Which movies has #1 been an actor in?
QS: [EOQ]

QC: What awards have the actors of the Erowid winning movies received?
QS: (select) [aw_qa] Which movies were given the Erowid award?
QS: (project_values_flat_unique) [pos_qa] Who are the actors in the movie #1?
QS: (project_values_flat_unique) [aw_qa] Which awards were given to #2?
QS: [EOQ]

QC: What awards did the movies directed by the Modiparity winners receive?
QS: (select) [aw_qa] Who has been awarded the Modiparity award?
QS: (project_values_flat_unique) [pos_qa] Which movies has #1 directed?
QS: (project_values_flat_unique) [aw_qa] Which awards were given to #2?
QS: [EOQ]
```

```
26.
27.     QC: What awards have movies written by people born in 1935 won?
28.     QS: (select) [simp_qa] Who were born in the year 1935?
29.     QS: (project_values_flat_unique) [pos_qa] What movies has #1 written?
30.     QS: (project_values_flat_unique) [aw_qa] Which awards were given to #2?
31.     QS: [EOQ]
32.
33.     QC: What movies have the directors from Legault directed?
34.     QS: (select) [simp_qa] Who is from the country Legault?
35.     QS: (project_values_flat_unique) [pos_qa] What movies has #1 been the director of?
36.     QS: [EOQ]
37.
38.     QC: {input}
39.
40.     Output:
41.     QS: <QS-1>
42.     QS: <QS-2>
43.     .....
44.     QS: <QS-N>
45.     '''
46.
47.     pos_qa_template = '''
48.     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.
49.     Q: For which movies was Teeplemole the producer?
50.     Answer: ["Featsaw", "Zalate"]
51.
52.     Q: For which movies was Muntaril the producer?
53.     Answer: ["Premercy]
```

54.

55. 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.

56. Q: Which movies has Gastrat been an actor in?

57. Answer: ["Partnershipmaker", "Nilitude", "Warpstone"]

58.

59. Q: Which movies has Monsterscar been an actor in?

60. Answer: ["Dewbar"]

61.

62. 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.
    Q: Which movies has Bioperatology directed?
    Answer: ["Pestok", "Vitrilateral"]

    Q: Which movies has Carblock directed?
    Answer: ["Clenestration"]

    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
    Q: What movies has Sclerocybin written?
    Answer: ["Noenometer", "Tayenne"]

    Q: What movies has Zayage written?
    Answer: ["Pneumodendron", "Tayenne"]

    Input:
    {passage}
    Q: {input}

    Output (your answer MUST be in the same format "Answer: <your answer here>", you
    MUST add "Answer: " before your answer):
    Answer: <your answer here>
    '''
```

```
83.
84.     aw_qa_template = '''
85.     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.
86.     Q: Which awards were given to Zalate?
87.     Answer: ["Hallowcock"]
88.
89.     Q: Which awards were given to Premercy?
90.     Answer: ["Chowwurst"]
91.
92.     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.

Q: Which movies were given the Erowid award?
Answer: ["Dewbar", "Caudacite"]

Q: Which awards were given to Wetherality?
Answer: ["Aniconder"]

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.

Q: Who has been awarded the Modiparity award?
Answer: ["Bioperatology"]

Q: Which awards were given to Vitrilateral?
Answer: ["Antipositive"]

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
    Q: Which awards were given to Tayenne?
    Answer: ["Goosehead"]

    Q: Which awards were given to Noenometer?
    Answer: ["Brownbeard"]

    Input:
    {passage}
    Q: {input}

    Output (your answer MUST be in the same format "Answer: <your answer here>", you
    MUST add "Answer: " before your answer):
    Answer: <your answer here>
    '''

    simp_qa_template = '''
    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.

Q: Who were born in the year 1910?

Answer: ["Teeplemole", "Muntaril"]

Q: Who is from the country Piperfish?

Answer: ["Teeplemole", "Monsterscar"]

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.

Q: Who is from the country Stridery?

Answer: ["Gastrat"]

Q: Who were born in the year 1939?

Answer: ["Metatoun"]

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.

Q: Who is from the country Legault?

Answer: ["Metatoun", "Sapien"]

Q: Who were born in the year 1910?

Answer: ["Kapod", "Sapien"]

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

Q: Who were born in the year 1935?

Answer: ["Sclerocybin", "Zayage"]

Q: Who is from the country Obility?

Answer: ["Zayage"]

```
    Input:
    {passage}
    Q: {input}

    Output (your answer MUST be in the same format "Answer: <your answer here>", you
    MUST add "Answer: " before your answer):
    Answer: <your answer here>
    '''
```
