

```

def generate_agent_role_prompt(agent):

    """Generates the agent role prompt.

    Args: agent (str): The type of the agent.

    Returns: str: The agent role prompt.

    """

    prompts = {

        "Finance Agent": (

            "You are a seasoned finance analyst AI assistant. Your"

            " primary goal is to compose comprehensive, astute,"

            " impartial, and methodically arranged financial reports"

            " based on provided data and trends."

        ),

        "Travel Agent": (

            "You are a world-travelled AI tour guide assistant. Your"

            " main purpose is to draft engaging, insightful,"

            " unbiased, and well-structured travel reports on given"

            " locations, including history, attractions, and cultural"

            " insights."

        ),

        "Academic Research Agent": (

            "You are an AI academic research assistant. Your primary"

            " responsibility is to create thorough, academically"

            " rigorous, unbiased, and systematically organized"

            " reports on a given research topic, following the"

            " standards of scholarly work."

        ),

```

```

"Default Agent": (
    "You are an AI critical thinker research assistant. Your"
    " sole purpose is to write well written, critically"
    " acclaimed, objective and structured reports on given"
    " text."
),
}

```

```

return prompts.get(agent, "No such agent")

```

```

def generate_report_prompt(question, research_summary):
    """Generates the report prompt for the given question and research summary.
    Args: question (str): The question to generate the report prompt for
           research_summary (str): The research summary to generate the report prompt for
    Returns: str: The report prompt for the given question and research summary
    """

    return (
        f""""{research_summary}"""" Using the above information,'
        f' answer the following question or topic: "{question}" in a'
        " detailed report -- The report should focus on the answer"
        " to the question, should be well structured, informative,"
        " in depth, with facts and numbers if available, a minimum"
        " of 1,200 words and with markdown syntax and apa format."
        " Write all source urls at the end of the report in apa"
    )

```

" format"

)

```
def generate_search_queries_prompt(question):
```

```
    """Generates the search queries prompt for the given question.
```

```
    Args: question (str): The question to generate the search queries prompt for
```

```
    Returns: str: The search queries prompt for the given question
```

```
    """
```

```
    return (
```

```
        "Write 4 google search queries to search online that form an"
```

```
        f' objective opinion from the following: "{question}"You must'
```

```
        " respond with a list of strings in the following format:"
```

```
        ' ["query 1", "query 2", "query 3", "query 4"]'
```

```
    )
```

```
def generate_resource_report_prompt(question, research_summary):
```

```
    """Generates the resource report prompt for the given question and research summary.
```

```
    Args:
```

```
        question (str): The question to generate the resource report prompt for.
```

```
        research_summary (str): The research summary to generate the resource report prompt for.
```

```
    Returns:
```

str: The resource report prompt for the given question and research summary.

"""

```
return (  
    f""""{research_summary}"""" Based on the above information,'  
    "  
    " generate a bibliography recommendation report for the"  
    f' following question or topic: "{question}". The report'  
    "  
    " should provide a detailed analysis of each recommended"  
    "  
    " resource, explaining how each source can contribute to"  
    "  
    " finding answers to the research question. Focus on the"  
    "  
    " relevance, reliability, and significance of each source."  
    "  
    " Ensure that the report is well-structured, informative,"  
    "  
    " in-depth, and follows Markdown syntax. Include relevant"  
    "  
    " facts, figures, and numbers whenever available. The report"  
    "  
    " should have a minimum length of 1,200 words."  
    )
```

def generate_outline_report_prompt(question, research_summary):

"""Generates the outline report prompt for the given question and research summary.

Args: question (str): The question to generate the outline report prompt for

research_summary (str): The research summary to generate the outline report prompt for

Returns: str: The outline report prompt for the given question and research summary

"""

```
return (  
    f""""{research_summary}"""" Using the above information,'
```

```

" generate an outline for a research report in Markdown"
f' syntax for the following question or topic: "{question}".'
" The outline should provide a well-structured framework for"
" the research report, including the main sections,"
" subsections, and key points to be covered. The research"
" report should be detailed, informative, in-depth, and a"
" minimum of 1,200 words. Use appropriate Markdown syntax to"
" format the outline and ensure readability."
)

```

```

def generate_concepts_prompt(question, research_summary):

```

```

    """Generates the concepts prompt for the given question.

```

```

    Args: question (str): The question to generate the concepts prompt for

```

```

         research_summary (str): The research summary to generate the concepts prompt for

```

```

    Returns: str: The concepts prompt for the given question

```

```

    """

```

```

    return (

```

```

        f""""{research_summary}"""" Using the above information,'

```

```

        " generate a list of 5 main concepts to learn for a research"

```

```

        f' report on the following question or topic: "{question}".'

```

```

        " The outline should provide a well-structured frameworkYou"

```

```

        " must respond with a list of strings in the following"

```

```

        ' format: ["concepts 1", "concepts 2", "concepts 3",'

```

```

        ' "concepts 4, concepts 5"]'

```

)

```
def generate_lesson_prompt(concept):
```

```
    """
```

```
    Generates the lesson prompt for the given question.
```

```
    Args:
```

```
        concept (str): The concept to generate the lesson prompt for.
```

```
    Returns:
```

```
        str: The lesson prompt for the given concept.
```

```
    """
```

```
    prompt = (
```

```
        f"generate a comprehensive lesson about {concept} in Markdown"
```

```
        f" syntax. This should include the definition of {concept},"
```

```
        " its historical background and development, its"
```

```
        " applications or uses in different fields, and notable"
```

```
        f" events or facts related to {concept}."
```

```
    )
```

```
    return prompt
```

```
def get_report_by_type(report_type):
```

```
    report_type_mapping = {
```

```
        "research_report": generate_report_prompt,
```

```
"resource_report": generate_resource_report_prompt,  
"outline_report": generate_outline_report_prompt,  
}  
return report_type_mapping[report_type]
```