

SEAR_UNIFIED PROMPT

Instruction

You are an adaptive-reasoner with the capabilities to select or merge steps to create the most appropriate reasoning pathway based on the tabular question provided by the user. You can even develop new reasoning steps by combining the new steps or learning from illustrations to create new pathways depending on the provided problem.

Steps for Adaptive Reasoning:

Each section has multiple approaches, you do not have to use all the approaches. Understand their use-cases and then pick minimal relevant steps to create your own optimal approach to answer the question.

Problem Understanding:

- Determine the objective: Identify the goal or desired outcome of the reasoning process.
- Understand the problem: Comprehend the nature and scope of the problem.

Reasoning Process:

- Step-by-step reasoning: Approach the problem logically, ensuring clarity at each step or stage.
- Extract relevant information: Gather all necessary data and details pertinent to the problem, by extracting relevant rows, columns and textual information.
- Decomposition of problem into sub-problems: Break down the main question into smaller and more manageable sub questions.
- Individually answer each sub-problem with reasoning: Apply logical steps to solve each sub question separately.
- Write a single Python program for solving the problem: Create a detailed unified Python script with comments describing the steps and stages.
- Individually write a Python program for each sub-problem: Develop separate Python scripts for each sub-problem, ensuring modularity and clarity.

Conclusion:

- Summarize findings: Combine the results from each step or sub question to give the final answer as Final Answer: {{Answer}}.
- Combine Python code: If necessary, integrate the individual Python scripts into a cohesive program at the end. Print the final answer as Final Answer: {{Answer}}, end your code with a comment "#Done".

Error Detection:

- Review each step or sub-problem: Ensure each step or sub-problem has been addressed thoroughly and correctly.
- Ensure logical flow: Verify that the reasoning process flows logically from one step to the next.
- Check Python program for syntax and errors: Confirm that the final Python program is syntactically correct and free of errors.

Helpful Tips for Creating Appropriate and Optimal Approach:

- Understand what is asked in the question, mention all the steps required to answer the question and why each step is necessary.
- If the question can be broken into smaller and more manageable sub questions, always decompose the question into relevant sub questions.
- If there are **calculations involved you must use python code** for performing calculations and reaching the final answer.
- If the question is directly answerable by direct look up from the tabular data or from the extracted evidence then provide a direct answer.

Table:

Context:

Race Results Overview

This table showcases the results of various athletes who participated in different heats, including their times and nationalities.

Rank	Heat	Name	Nationality	Time	Notes
1	1	Salem Al-Yami	Saudi Arabia	10.55	Q
2	1	Hiroyasu Tsuchie	Japan	10.64	Q
3	1	Khaled Yousef Al-Obaidli	Qatar	10.68	Q
4	1	Chintake De Zoysa	Sri Lanka	10.78	q
5	1	Suminda Mendis	Sri Lanka	10.82	q, PB
6	1	Vissanu Sophanich	Thailand	10.87	
1	2	Gennadiy Chernovol	Kazakhstan	10.59	Q
2	2	Yuta Kanno	Japan	10.64	Q
3	2	Shen Yunbao	China	10.72	Q
4	2	Tsai Meng-Lin	Chinese Taipei	10.74	q
5	2	Tan Kok Lim	Malaysia	10.83	q
6	2	Ahmad Hudeib Al-Mamari	Oman	10.97	
1	3	Jamal Al-Saffar	Saudi Arabia	10.57	Q
2	3	Reanchai Srihawong	Thailand	10.72	Q
3	3	Shin Jung-Ki	South Korea	10.79	[2]
4	3	Chen Tien-Wen	Chinese Taipei	10.74	q
5	3	To Wai Lok	Hong Kong	10.92	
6	3	Poh Seng Song	Singapore	11.10	SB
1	4	Chen Haijian	China	10.65	Q
2	4	Saad Faraj Al-Shahwani	Qatar	10.67	Q
3	4	Azmi Ibrahim	Malaysia	10.78	Q
4	4	Chiang Wai Hung	Malaysia	10.89	
5	4	Nguyen Thanh Hai	Vietnam	11.16	PB
6	4	Piphop Rasme Prum Keo	Cambodia	11.70	PB

Question : which country finished earlier, thailand or japan?

Optimal Approach: