AI Agent Assessment Objective: To evaluate your ability to design and implement a basic AI agent that can mimic the persona and behavior of a Formula One racer, suitable for social media interactions. This assessment tests your understanding of natural language processing, potentially large language models, and agent design principles. **Task:** You are required to create a Python-based AI agent that can:

1.**Speak (Generate Text):** Generate text messages, comments, or social media posts inthe style of a Formula One racer. This includes using relevant vocabulary, expressingtypical racer sentiments (e.g., focus, determination, excitement, disappointment), andpotentially referencing racing events or team dynamics.

2.**Act (Perform Actions - Simulated):** Simulate basic social media actions a racer mighttake, such as:

○"Replying" to a comment with a generated text.

○"Posting" a new status update.

○"Liking" a post (no actual liking needed, just simulate the action).

○"Mentioning" a teammate or competitor in a post.

3.**Think (Contextual Awareness - Basic):** Maintain a very basic awareness of asimulated "race weekend" context. This could involve:

○Knowing the current "stage" (e.g., practice, qualifying, race).

○Remembering a recent "result" (e.g., good, bad, DNF).

○Tailoring the generated text based on this context.

Technical Requirements:

●The agent should be implemented in Python.

●The application must be in a docker container with all the code required to create, runand interact with the container.

●You are encouraged to leverage relevant NLP libraries (e.g., NLTK, spaCy,Transformers) or even a small pre-trained language model if you deem it beneficial fortext generation (though not strictly required for a basic implementation).

- The code should be well-structured, readable, and include comments explaining thelogic.
- The agent should have a clear interface (e.g., functions or a class with methods) totrigger speaking, acting, and updating its internal "thinking" state.

Deliverables:

1. A Python script (f1_agent.py) containing the implementation of your AI agent.
2. A README.md file in your repository that includes:
    - ○ A brief description of your agent and its capabilities.
    - ○ Instructions on how to run the script.
    - ○ Examples of the agent's output for different scenarios (e.g., after a win, after a difficult race, during practice).
    - ○ A brief explanation of your design choices and any challenges you encountered.

Example Scenarios:

Your agent should be able to generate outputs similar to the following (depending on its internal "thinking" state):

- **After a Win:** "YES! What a race! Huge thanks to the team for the amazing car. We pushed hard and it paid off. #Winner #Team[TeamName] #[RaceName]"
- **After a Difficult Race:** "Not the result we wanted today. Gave it my all out there, but things didn't go our way. We'll analyze and come back stronger next time. Thanks for the support. #NeverGiveUp #[RaceName]"
- **During Practice:** "Getting some good laps in during FP2. Feeling comfortable with the car. Let's keep pushing! #[RaceName] #FP2"
- **Replying to a fan comment:** *(Fan: "Great drive today, [RacerName]!")* Agent Reply: "Thanks for the support! Every cheer makes a difference. ⍰⍰"

Evaluation Criteria:

Your submission will be evaluated based on the following:

- **Persona Accuracy:** How well the generated text and simulated actions capture the essence of a Formula One racer's personality and behavior.

- **Technical Implementation:** The quality of your Python code, including structure, readability, and efficiency.

- **Contextual Awareness:** The agent's ability to adapt its output based on the simulated "race weekend" context.

- **Creativity and Effort:** The level of creativity and effort demonstrated in the design and implementation of the agent.

- **Documentation:** The clarity and completeness of your README.md file.

Submission Instructions:

1. Create a public repository on your GitHub account named f1_racer_ai_agent.

2. Commit your f1_agent.py and README.md files to this repository.

3. Submit the link to your GitHub repository as your assessment submission.

We look forward to reviewing your creative and technical skills in building this basic AI agent! Good luck!