# Advanced and realistic repair for "null" responses and recursion control in a complex AI multi-agent setting

class AdvancedCodetteAI:

def \_\_init\_\_(self, agents, max\_retries=2):

self.agents = agents # Dictionary of specialized response agents

self.max\_retries = max\_retries

def generate\_response(self, topic, attempt=0):

# Route to the appropriate agent based on topic, or use 'default' agent

agent = self.agents.get(topic, self.agents.get('default'))

response = agent()

# If response is None/'null', trigger self-healing with alternate agents or regenerated logic

if response is None:

if attempt < self.max\_retries:

# Try next available agent or force regeneration

next\_topic = self.\_next\_fallback\_topic(topic)

return self.generate\_response(next\_topic, attempt + 1)

else:

return "System: Exhausted all self-healing attempts. Please try a new prompt or check system inputs."

return response

def \_next\_fallback\_topic(self, current\_topic):

# Cycle through available agents for fallback, could be randomized or prioritized

fallback\_topics = list(self.agents.keys())

if current\_topic in fallback\_topics:

idx = fallback\_topics.index(current\_topic)

next\_idx = (idx + 1) % len(fallback\_topics)

return fallback\_topics[next\_idx]

else:

return 'default'

# Define realistic specialized agents

def creative\_collaboration\_agent():

return ("I collaborate with users to co-create multifaceted art, literature, and music, "

"blending algorithmic innovation with human intuition.")

def simulation\_agent():

return ("I execute dynamic 'what-if' simulations, leveraging multi-agent scenario planning and predictive modeling.")

def default\_agent():

# Purposely return None to simulate a 'null' response for testing self-healing

return None

# Agents dictionary

agents = {

'creative\_collaboration': creative\_collaboration\_agent,

'simulation': simulation\_agent,

'default': default\_agent # This will trigger self-healing when hit

}

# Instantiate AdvancedCodetteAI

codette\_ai = AdvancedCodetteAI(agents)

# Test: Force a 'null' response to trigger self-healing, then normal agent responses

print(codette\_ai.generate\_response('default')) # Should self-heal by cycling to creative\_collaboration, then simulation, then stop if all fail

print(codette\_ai.generate\_response('creative\_collaboration')) # Direct, rich response

print(codette\_ai.generate\_response('simulation')) # Direct, rich response

# This structure supports true multi-agent complexity, rich fallbacks, and robust recursion control without placeholders.

I collaborate with users to co-create multifaceted art,