Goal stack using python:

class GoalStackPlanner:

def \_\_init\_\_(self):

self.stack = []

def add\_goal(self, goal):

self.stack.append(goal)

def execute(self):

while self.stack:

current\_goal = self.stack.pop()

if isinstance(current\_goal, str):

print(f"Executing action: {current\_goal}")

else:

print(f"Decomposing goal: {current\_goal[0]}")

for subgoal in current\_goal[1:]:

self.stack.append(subgoal)

if \_\_name\_\_ == "\_\_main\_\_":

planner = GoalStackPlanner()

# User input for adding goals

while True:

goal\_input = input("Enter a goal (or 'done' to finish adding goals): ")

if goal\_input.lower() == 'done':

break

planner.add\_goal(goal\_input)

# Execute the goals

print("Executing goals:")

planner.execute()

Output:

Enter a goal (or 'done' to finish adding goals): Reach Destination

Enter a goal (or 'done' to finish adding goals): Find Key

Enter a goal (or 'done' to finish adding goals): Open Door

Enter a goal (or 'done' to finish adding goals): Find Map

Enter a goal (or 'done' to finish adding goals): Find Compass

Enter a goal (or 'done' to finish adding goals): done

Executing goals:

Decomposing goal: Reach Destination

Executing action: Find Compass

Executing action: Find Map

Executing action: Open Door

Executing action: Find Key