**Riphah International University**

**Artificial Intelligence (AI)**

**Lab 3**

**Submitted by:** Muhammad Qasim

**Sap ID:** 37137

**Section:** BSCS-6A

**Submitted To:** Mahjabeen

**Riphah School of Computing & Innovation Faculty of Computing Riphah International University, Lahore 2023**

**Lab**

def print\_state(state):

    print("Left:", state["left"])

    print("Right:", state["right"])

    print("Boat:", state["boat"])

    print()

def move(state):

    print\_state(state)

    print("Available actions:")

    for i, action in enumerate(actions, 1):

        print(f"{i}. {action}")

    while True:  # Keep asking until a valid input is provided

        choice = input("Choose an action (1-4): ")

        if choice.isdigit() and 1 <= int(choice) <= 4:

            break

        else:

            print("Invalid input. Please enter a number between 1 and 4.")

    action = actions[int(choice) - 1]

    if state["boat"] == "left":

        for item in action.split(", "):

            if item in state["left"]:

                state["left"].remove(item)

                state["right"].add(item)

    else:

        for item in action.split(", "):

            if item in state["right"]:

                state["right"].remove(item)

                state["left"].add(item)

    state["boat"] = "right" if state["boat"] == "left" else "left"

def is\_goal\_state(state):

    return state == goal\_state

def is\_valid\_move(state, action):

    new\_state = state.copy()

    if new\_state["boat"] == "left":

        for item in action.split(", "):

            if item not in new\_state["left"]:

                return False

            new\_state["left"].remove(item)

            new\_state["right"].add(item)

    else:

        for item in action.split(", "):

            if item not in new\_state["right"]:

                return False

            new\_state["right"].remove(item)

            new\_state["left"].add(item)

    return is\_valid(new\_state) and not goat\_eats\_corn(new\_state) and not goat\_eats\_wolf(new\_state)

def goat\_eats\_corn(state):

    return "goat" in state["left"] and "corn" in state["left"] and "farmer" not in state["left"]

def goat\_eats\_wolf(state):

    return "goat" in state["left"] and "wolf" in state["left"] and "farmer" not in state["left"]

initial\_state = {"left": {"farmer", "goat", "corn", "wolf"}, "right": set(), "boat": "left"}

goal\_state = {"left": set(), "right": {"farmer", "goat", "corn", "wolf"}, "boat": "right"}

actions = [

    "farmer",

    "farmer, goat",

    "farmer, corn",

    "farmer, wolf"

]

current\_state = initial\_state.copy()

while not is\_goal\_state(current\_state):

    move(current\_state)

    if goat\_eats\_corn(current\_state):

        print("Oops! The goat ate the corn. You lose!")

        break

    elif goat\_eats\_wolf(current\_state):

        print("Oops! The goat ate the wolf. You lose!")

        break

if is\_goal\_state(current\_state):

    print("Congratulations! You have successfully moved all characters to the right side.")

OutPut:

