1. def minimax(depth, idx, is\_max, values, alpha, beta):

if depth == 3:

return values[idx]

if is\_max:

best = float('-inf')

for i in range(2):

best = max(best, minimax(depth + 1, idx \* 2 + i, False, values, alpha, beta))

alpha = max(alpha, best)

if beta <= alpha:

break

return best

else:

best = float('inf')

for i in range(2):

best = min(best, minimax(depth + 1, idx \* 2 + i, True, values, alpha, beta))

beta = min(beta, best)

if beta <= alpha:

break

return best

values = [3, 5, 6, 1, 2, 0, -1, 4]

print("Optimal value:", minimax(0, 0, True, values, float('-inf'), float('inf')))

2. facts = ["Croaks", "Eat Flies", "Shrimps", "Sings"]

animals = ["Frog", "Canary"]

colors = ["Green", "Yellow"]

def main():

print("----- Forward Chaining -----")

for i, fact in enumerate(facts, 1):

print(f"{i}. {fact}")

try:

x = int(input("\nSelect one: "))

if x not in range(1, 5):

print("Invalid option.")

return

print(f"\nX is {facts[x - 1]}")

print("1. Green\n2. Yellow")

k = int(input("Select color option: "))

if x in [1, 2] and k == 1:

print(f"\nYes, it is {animals[0]} and its color is {colors[0]}")

elif x in [3, 4] and k == 2:

print(f"\nYes, it is {animals[1]} and its color is {colors[1]}")

else:

print("\nNo matching rule found.")

except ValueError:

print("Please enter a valid number.")

main()