1.

import numpy as np

import matplotlib.pyplot as plt

# Given probabilities

P\_D = 0.001 # Prior: P(Disease)

P\_not\_D = 1 - P\_D

P\_Pos\_given\_D = 0.99 # Sensitivity

P\_Pos\_given\_not\_D = 0.02 # False Positive Rate

# Simulate repeated positive tests

n\_tests = 10

posterior\_history = [P\_D] # Start with prior

current\_posterior = P\_D

for i in range(n\_tests):

P\_Pos = P\_Pos\_given\_D \* current\_posterior + P\_Pos\_given\_not\_D \* (1 - current\_posterior)

current\_posterior = (P\_Pos\_given\_D \* current\_posterior) / P\_Pos

posterior\_history.append(current\_posterior)

# Plotting

plt.figure(figsize=(8, 5))

plt.plot(range(n\_tests + 1), posterior\_history, marker='o', linestyle='-', color='purple')

plt.xlabel('Number of Positive Tests')

plt.ylabel('P(Disease | Positive Tests)')

plt.title('Bayesian Update After Repeated Positive Tests')

plt.grid(True)

plt.ylim(0, 1)

plt.xticks(range(n\_tests + 1))

plt.show()

2.

import numpy as np

import matplotlib.pyplot as plt

# Given probabilities

P\_D = 0.001 # Prior: P(Disease)

P\_not\_D = 1 - P\_D

P\_Pos\_given\_D = 0.99 # Sensitivity

P\_Pos\_given\_not\_D = 0.02 # False Positive Rate

# Simulate repeated positive tests

n\_tests = 10

posterior\_history = [P\_D] # Start with prior

current\_posterior = P\_D

for i in range(n\_tests):

P\_Pos = P\_Pos\_given\_D \* current\_posterior + P\_Pos\_given\_not\_D \* (1 - current\_posterior)

current\_posterior = (P\_Pos\_given\_D \* current\_posterior) / P\_Pos

posterior\_history.append(current\_posterior)

# Plotting

plt.figure(figsize=(8, 5))

plt.plot(range(n\_tests + 1), posterior\_history, marker='o', linestyle='-', color='purple')

plt.xlabel('Number of Positive Tests')

plt.ylabel('P(Disease | Positive Tests)')

plt.title('Bayesian Update After Repeated Positive Tests')

plt.grid(True)

plt.ylim(0, 1)

plt.xticks(range(n\_tests + 1))

plt.show()

3.

import numpy as np

import matplotlib.pyplot as plt

# Initial prior: Probability other parent is a carrier

prior = 0.1

P\_red\_given\_carrier = 0.25

P\_red\_given\_not\_carrier = 0.0

# Simulate belief update after multiple red-haired children

n\_children = 5

posterior\_history = [prior]

current\_posterior = prior

for i in range(n\_children):

# Total probability of red hair

P\_red = P\_red\_given\_carrier \* current\_posterior + P\_red\_given\_not\_carrier \* (1 - current\_posterior)

# Bayes' update

current\_posterior = (P\_red\_given\_carrier \* current\_posterior) / P\_red

posterior\_history.append(current\_posterior)

# Plotting the belief update

plt.figure(figsize=(8, 5))

plt.plot(range(n\_children + 1), posterior\_history, marker='o', linestyle='-', color='crimson')

plt.xlabel('Number of Red-Haired Children')

plt.ylabel('P(Other Parent is Carrier)')

plt.title('Belief Update: Probability Other Parent is a Carrier')

plt.ylim(0, 1.05)

plt.grid(True)

plt.xticks(range(n\_children + 1))

plt.show()