```python
import numpy as np
import matplotlib.pyplot as plt
from math import comb
from scipy import stats

def log_lik(theta, x, y, n):
    # Computes the log-likelihood of the data assuming a Binomial distribution.
    alpha, beta = theta
    p = 1 / (1 + np.exp(-(alpha + beta * x)))
    ll = np.sum(stats.binom.logpmf(y, n, p))
    return ll

def lik(theta, x, y, n):
    # Computes the likelihood by exponentiating the log-likelihood.
    return np.exp(log_lik(theta, x, y, n))

def neg_log_lik(theta, x, y, n):
    # Computes the negative log-likelihood, used for minimization.
    return -log_lik(theta, x, y, n)

def neg_lik(theta, x, y, n):
    # Computes the negative likelihood.
    return -lik(theta, x, y, n)

def log_prior(theta):
    # Computes the log-prior probability assuming independent Gaussians (constants omitted).
    alpha, beta = theta
    lp_alpha = - (alpha**2) / (2 * 2**2)  # sigma=2
    lp_beta =  - (beta**2)  / (2 * 1**2)  # sigma=1
    return lp_alpha + lp_beta

def log_posterior(theta, x, y, n):
    # Computes the unnormalized log-posterior (sum of log-likelihood and log-prior).
    return log_lik(theta, x,y,n) + log_prior(theta)

def neg_log_posterior(theta,x,y,n):
    # Computes the negative log-posterior, used for MAP estimation.
    return -log_posterior(theta, x,y,n)
```