Let us assume a two-class classification problem where each class is modeled by a 2D Gaussian distribution $G(\mu_1, \Sigma_1)$ and $G(\mu_2, \Sigma_2)$.

1. Generate 10,000 samples from each 2D Gaussian distribution using the following parameters (i.e., each sample $(x,y)$ can be thought as a feature vector):

   \[
   \begin{align*}
   \mu_1 &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}, & \Sigma_1 &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \\
   \mu_2 &= \begin{bmatrix} 6 \\ 6 \end{bmatrix}, & \Sigma_2 &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}
   \end{align*}
   \]

   **Note:** this is not the same as sampling the 2D Gaussian functions; see “Generating Gaussian Random Numbers” on the course’s webpage for more information on how to generate the samples using the Box-Muller transformation. A link to C code has been provided. Since the code generates samples for 1D distributions, you would need to call the function twice to get a 2D sample $(x, y)$; use the $x$-mean, $x$-variance for the $x$ sample and the $y$-mean, $y$-variance for the $y$ sample.

   a. Assuming $P(\omega_1) = P(\omega_2)$
      i. Design a Bayes classifier for minimum error.
      ii. Plot the Bayes decision boundary **together** with the generated samples to better visualize and interpret the classification results.
      iii. Report (1) the number of misclassified samples for each class separately and (2) the total number of misclassified samples.
      iv. Plot the Chernoff bound as a function of $\beta$ and find the optimum $\beta$ for the minimum.
      v. Calculate the Bhattacharyya bound. Is it close to the experimental error?

   b. Repeat part (a) for $P(\omega_1) = 0.2$ and $P(\omega_2) = 0.8$.

2. Repeat (1) using the following parameters:

   \[
   \begin{align*}
   \mu_1 &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}, & \Sigma_1 &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \\
   \mu_2 &= \begin{bmatrix} 6 \\ 6 \end{bmatrix}, & \Sigma_2 &= \begin{bmatrix} 4 & 0 \\ 0 & 8 \end{bmatrix}
   \end{align*}
   \]

3. For part 2, use the minimum-distance classifier to classify the samples and compare your results (i.e., misclassified samples) with the Bayes classifier from part 2.