1. **[10 marks]** Consider the problem of finding the greatest common divisor of two positive integers—the largest positive integer that divides both of them without a remainder. Described in his *Elements* (c. 300 BC) and being one of the oldest algorithms, Euclid's algorithm (a modern version) solves the problem as follows:

   > Keep replacing the larger of the two numbers by its remainder when divided by the smaller of the two, until the remainder becomes zero. Then the smaller number is the solution.

   Implement this algorithm using a recursive function. Make sure that the recycling rule works, i.e., the function can process two vectors of unequal lengths. Inside your function, you do not have to check the validity of the input values and may simply assume that they are positive integers. Demonstrate that your function works properly in the following two cases:

   ```
   > gcd(6, 15)
   > gcd(6:10, 15:25)
   ```

2. **[20 marks]** In this question, let us investigate a classification problem that has many more variables than observations, i.e., a $p \gg n$ problem. As required in parts (a) and (b) below, two classification methods (slightly simplified) are to be implemented and used.

   Modern technology such as DNA microarrays has made it feasible to measure the expression levels of large numbers of genes simultaneously for a sample (or subject/observation). On the course web site, the following three files are available (data source: Golub et al. (1999), *Science* 286, 531–536):

`training.csv`	-	Training data set needed for building a classifier. It contains the expression levels of $p = 7129$ genes for $n = 38$ samples, along with the gene accession numbers (or gene IDs).
`test.csv`	-	Independent test data set that can be used for examining the performance of a classifier. It contains the expression levels of the same 7129 genes for 34 new samples, where the genes, as indicated by their accession numbers, are given in the same order as in the training set.
`samples.txt`	-	Class labels of the 72 ($= 38 + 34$) samples. The class of a sample is one of two types of tumor, either acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML).

   Researchers are interested in being able to predict the type of tumor from the expression levels of genes. For a two-class problem like this, one can label/classify a new observation $\mathbf{x} = (x_1, \ldots, x_p)^\top$ as class 1 (or ALL) if

   $$f_1(\mathbf{x}) > f_2(\mathbf{x}),$$

or class 2 (or AML) if otherwise, where f_1 and f_2 are, respectively, the density functions of the observations in the two classes. If the variables x_j are treated as being independent and normally distributed, the above classification criterion is equivalent to using the inequality given by

$$\sum_{j=1}^{p} \log\{\phi(x_j; \mu_{1j}, \sigma_j)\} > \sum_{j=1}^{p} \log\{\phi(x_j; \mu_{2j}, \sigma_j)\}, \tag{1}$$

where $\phi(\cdot; \mu, \sigma)$ denotes the univariate normal density function with mean μ and standard deviation σ, μ_{kj} $(k = 1, 2)$ the mean of the expression levels of gene j for class k and σ_j the corresponding within-class standard deviation. The two classification methods below differ in their estimates of μ_{kj}, where σ_j are replaced with their pooled estimates $\hat{\sigma}_j$ from the training data (check the two-sample t test example in the lecture slides for how they should be computed).

Note that as typical for gene data, the genes are given along the rows but should be treated as variables, while the samples are along the columns but should be treated as observations, in both the training and test sets.

In your code, you should avoid using loops as much as you can.

(a) [10 marks] The *Näive Bayes* method simply uses the sample mean $\hat{\mu}_{kj}$ in place of the unknown true mean μ_{kj}. Implement this classifier and apply it to the test set. Provide the predicted class labels of the test samples and find out how many test samples are misclassified.

(b) [10 marks] A problem with the Näive Bayes method is that it includes all variables in its classifier, but most variables (genes here) are likely irrelevant to classification. A soft thresholding method known as *Nearest Shrunken Centroids* (Tibshirani et al. (2002), *PNAS* 99, 6567–6572) provides a better solution. It finds and uses only a subset of variables in its classifier and yet often gives more accurate predictions. The method works as follows. Let

$$d_{kj} = \frac{\hat{\mu}_{kj} - \hat{\mu}_j}{m_k \hat{\sigma}_j},$$

where $\hat{\mu}_j$ is the overall mean for gene j and $m_k = \sqrt{1/n_k - 1/n}$, n_k being the number of samples of class k. For a properly-chosen threshold value $\lambda > 0$, d_{kj} is shrunken towards 0 using

$$d'_{kj} = \text{sign}(d_{kj})(|d_{kj}| - \lambda)_+,$$

where "+" means positive part ($a_+ = a$ if $a > 0$, and $= 0$ otherwise). The shrunken versions of $\hat{\mu}_{kj}$ are then obtained by reversing the above transformation as follows:

$$\hat{\mu}'_{kj} = \hat{\mu}_j + m_k \hat{\sigma}_j d'_{kj}.$$

The classifier uses $\hat{\mu}'_{kj}$ in (1) in place of μ_{kj}. Note that for any j, if $d'_{1j} = d'_{2j} = 0$, then $\hat{\mu}'_{1j} = \hat{\mu}'_{2j} = \hat{\mu}_j$, which means that gene j has no predictive power and can be dropped from the classification criterion (1).

Implement the Nearest Shrunken Centroids method and apply it to the test set using $\lambda = 6$. Provide the predicted class labels of the test samples and find out how many test samples are misclassified. Print the accession numbers (IDs) of the genes that are deemed having predictive power by the method.

[An appropriate value of λ can be determined solely from the training set, via a data resampling method such as cross-validation. This is however beyond the scope of this course.]

3. **[20 marks]** The ABO locus used for blood types incorporates the three alleles A, B, and O and exhibits the four observable phenotypes A, B, AB and O. Since alleles A and B are genetically dominant to allele O, this results that, for example, both genotypes A/O and A/A exhibit the same phenotype A. From the observed numbers of people with blood types A, B, AB and O, respectively, one can estimate the proportions of alleles A, B and O in a population via the maximum likelihood method, which makes use of the Hardy-Weinberg law of population genetics.

   Suppose that a random sample of a human population is of size $n = 505$ and consists of $n_A = 172$, $n_B = 43$, $n_{AB} = 11$ and $n_O = 279$ people for phenotypes A, B, AB and O, respectively. The log-likelihood function is given by

$$l(p_A, p_B, p_O) = n_A \log(p_A^2 + 2p_A p_O) + n_B \log(p_B^2 + 2p_B p_O) + n_{AB} \log(2p_A p_B) + n_O \log(p_O^2),$$

   where $p_A, p_B, p_O \geq 0$ are the proportions of alleles A, B and O in the population, satisfying $p_A + p_B + p_O = 1$.

   (a) [10 marks] Use the `optim()` function to find the maximum likelihood estimates of p_A, p_B and p_O. You should use the BFGS (or L-BFGS-B) method, without providing the derivatives of the log-likelihood function.

   (b) [5 marks] Re-do part (a), with the derivatives of the log-likelihood function provided to `optim()`.

   (c) [5 marks] Provide a contour plot of the log-likelihood function, in terms of p_A and p_B only, where $p_O = 1 - p_A - p_B$.