Question 1

(a) We first use integration by parts. Write $\int_a^b h''(x)g''(x)dx = [h'(x)g''(x)]_a^b - \int_a^b h'(x)g'''(x)dx$. Now, note that since g is a natural cubic spline, it is linear outside the interval (x_1, x_N); hence, we have $g''(a) = g''(b) = 0$ and so our integral reduces to $-\int_a^b h'(x)g'''(x)dx$; since, as mentioned, g''' is zero outside of (x_1, x_N) we may write this integral as $-\int_{x_1}^{x_N} h'(x)g'''(x)dx$. Now, since g is a piecewise cubic function, g''' will be piecewise constant over (x_1, x_N); that is, g''' is constant over each interval $[x_j, x_{j+1})$. Thus, we can write $-\int_{x_1}^{x_N} h'(x)g'''(x)dx = -\sum_{j=1}^{N-1} \int_{x_j}^{x_{j+1}} g'''(x)h'(x)dx$.

Since g''' is right-continuous, we may write this as $-\sum_{j=1}^{N-1} \int_{x_j}^{x_{j+1}} g'''(x_j^+)h'(x)dx$

$= -\sum_{j=1}^{N-1} g'''(x_j^+) \int_{x_j}^{x_{j+1}} h'(x)dx = -\sum_{j=1}^{N-1} g'''(x_j^+)\left(h(x_{j+1}) - h(x_j)\right)$ as desired. Further, observe that since $g(x) = \tilde{g}(x)$ at the knots, $h(\cdot)$ is identically zero at every knot x_j. Thus, our sum is zero as desired.

(b) We have $\int_a^b (\tilde{g}''(t))^2 dt = \int_a^b (g''(t) + h''(t))^2 dt = \int_a^b \left((g''(t))^2 + g''(t)h''(t) + (h''(t))^2\right) dt$ $= \int_a^b \left((g''(t))^2 + (h''(t))^2\right) dt + 2\int_a^b g''(t)h''(t)dt$. Observe that by part (a), this second integral is zero; hence, we have $\int_a^b \left[(g''(t))^2 + (h''(t))^2\right] dt \geq \int_a^b (g''(t))^2 dt$. (since $(h''(t))^2 \geq 0 \forall t.$). Equality only holds if $\int_a^b (h''(t))^2 dt = 0$; but this can only happen if h'' is zero a.e. in (a, b). However, h is continuous and so if h'' is zero a.e. in (a, b) then h'' must be identically zero in $[a, b]$. The only way for h to include a cubic term (since $h = \tilde{g} - g$) and satisfy this property is for h to be identically zero in $[a, b]$.

(c) Holding λ fixed, given any function f there exists some natural cubic spline g with the same mean square error (i.e. $g(x_i) = f(x_i)$ for all the sample data points x_i). Hence, we consider $\lambda\int_a^b (f''(t))^2 dt$ versus $\lambda\int_a^b (g''(t))^2 dt$; however, from part (b) we know that for all functions f which interpolate the data points, we have $\int_a^b (f''(t))^2 dt \geq \int_a^b (g''(t))^2 dt$ with equality iff $f''(t) = g''(t)$. Hence, this natural cubic spline $g''(t)$ must be the minimizer.

Question 2

For k-nearest-neighbors, we consider only 1-, 7-, and 15-nearest neighbors. We carry out LDA on the entire data set; the process is straightforward, as we simply use the built-in LDA package in R. For QDA, we cannot carry it out on the entire data set, as several of the estimated covariance matrices are singular; in order to use QDA, we must first carry out some type of dimension reduction (in this case, PCA). We carry out PCA; we observe that the prediction errors are essentially identical for both the 50- and 100- component case, and so for computational simplicity we consider only the first 50 principal components of our new matrix (we could cross-validate here if need be for the optimal number of dimensions; but in this case, it appears that it hardly makes a difference given a sufficiently large number of principal components). Reducing the number of regressors from 256 to 50 is exactly the purpose of PCA; almost all the weight is placed upon these components, and so we may treat the other components more or less as noise variables. For logistic regression, we use the multinom package in R (from the class Design). For reduced-rank LDA, we first project the data onto a 10-dimensional subspace by virtue of linear regression; that is, if we let X be the input variable, we define $\hat{Y} = X(X^TX)^{-1}X^TY = X\hat{B}$ (where Y is the indicator response matrix). Doing this allows us to project our data on a subspace of dimension 10; we then compute the eigen-decomposition of $\hat{Y}Y$ by virtue of PCA. We cross-validate on the training set; using PCA allows us to choose how many components we wish to use each time then measure the error. The optimal number of components (i.e. that corresponding to the lowest average prediction error) will be the number of components we use when carrying out analyses upon the actual test set. It is interesting to note that although this greatly simplifies the LDA computations, the cross-validation step itself is computationally intensive (although still faster than discriminant analysis on the full data set). We note that by Exercise 4.3 of the textbook, LDA using all the regressors $\hat{Y}$ is identical to LDA in the original space (although, given that we have more than two classes, it is not equivalent to OLS). Cross-validating yields the optimal number of parameters as 9 (i.e. one less than the 10 parameters in our original reduction). Hence, we take the first nine components of the eigenvector obtained from PCA; these are the components with the most weight (i.e. most proportion of total variance). Carrying out the requisite computations, we obtain the following training and test errors:

Method	Test Error	Training Error
1-Nearest-Neighbor	3.79%	0%
7-Nearest-Neighbor	3.24%	1.22%
15-Nearest-Neighbor	3.99%	1.92%
LDA	11.46%	6.20%
QDA	6.52%	1.77%
Reduced-Rank LDA	11.46%	6.20%
Logistic Regression	9.49%	0%

From these results, we see that nearest-neighbour methods again perform better to discriminant analysis (LDA, QDA, and reduced-rank LDA) as well as logistic regression. This is likely again due to the nature of the problem and possibly our metric for error when using cross-validation; we have a discrete classification problem, and so when we cross-validate we use misclassification error rather than mean squared error. We note that in general, linear methods do not perform as well as nearest-neighbors for this classification problem; given that the Hastie/Tibshirani/Friedman website states that a 2.5 percent testing error is “excellent”, it is likely that the optimal method is much more complicated than linear. There is also the Gaussian assumption inherent in all of these methods (except nearest-neighbors); in the context of grayscale images this assumption may be a strong one, and as such one that causes LDA, QDA, reduced-rank LDA, and logistic regression to perform decently worse than nearest-neighbor methods. (We also note that as k grows, k-nearest-neighbors converges to ordinary least squares; this is why 7-nearest neighbors outperforms 15-nearest-neighbors in terms of both training and test error rate).
We also note that (as expected), the reduced-rank LDA and LDA methods give identical results to working precision; although the reduced-rank LDA used 9 eigenvectors as the basis for the

low-dimension subspace and not all 10 eigenvectors (in which case the reduced-rank LDA and regular LDA would be guaranteed to give identical results), the proportion of variance on the tenth is small enough that it is more or less irrelevant. Of particular interest is the superiority of QDA to LDA, logistic regression, and reduced-rank LDA; this is likely due to the aforementioned Gaussian assumption. QDA relaxes the assumption of a common covariance matrix, and instead allows for within-class variability. Given the nature of these images, this imposes fewer parameter restrictions upon the model and seems more reasonable than assuming a single variance across all groups.

Question 3

(a) In both cases (and in part (b) as well), we split the data into training and testing sets simply by cutting down the middle; i.e. we train on the first half of the observed data and test on the second half. Plotting this data and taking means of variables yields almost-identical results; hence, the data are not stratified and we may divide the data set this way without worrying about some kind of underlying structure in the ordering of the data. When carrying out raw logistic regression on the AA/AO data only, we obtain a 7.92% training error and a 24.71% testing error; when regularizing (using twelve uniformly spaced knots, as in the book) we instead obtain a 20.03% training error and an 18.97% testing error. Note that these don't precisely match up with the book, due to randomness/difference in the selection of training and testing sets. We do notice, however, the same trend as in the book; that is, imposing the natural cubic spline restriction causes our training error to increase, in this case quite substantially, but our testing error to decrease dramatically. This is likely due to the forced smoothness of the new estimator caused by the restriction; as seen in the book, this restriction greatly decreases the noisiness of the logistic regression fit. Our errors are likely high due to the simplicity of the model; although linear logistic regression in this case is probably better than QDA, the fact that many data points are highly correlated is something that logistic regression does not fully take into account.
When carrying out raw logistic regression on the full data set, we obtain a 1.42% training error and a 15.04% tests error; when carrying out regularized logistic regression on the full data set, we instead obtain a 12.90% training error and a 14.62% testing error. It is interesting to note that in this case, both errors are lower than in the AA-AO case; this is expected, because we have much more data to train on. It is perhaps a bit surprising that the change in test error between regularized and raw for the whole data set is minimal, whereas the change in training error is quite substantial; in this case, the results seem to indicate that the regularization over the whole data set perhaps serves to overidentify the model and as such hurts rather than helps. The imposition of the natural cubic spline basis likely creates too many moment conditions; as such, the raw logistic regression model, though by no means amazingly accurate, seems the better choice.

(b) When carrying out raw QDA, we obtain a 14.46% training error and a 29.02% testing error. When regularizing, for our five choices, we take 5, 6, 7, 8, and 9 knots; in each case, we space the knots uniformly across the data. Using tenfold cross-validation, we find 8 knots (note that $p+1$ degrees of freedom correspond to p knots)to be the optimal number out of these five. We therefore train and test using natural cubic splines with 8 knots. Under this new regularized model, we obtain a training error of 15.21% and a testing error of 17.66%. It is interesting to note that under regularization, our training error is actually slightly higher than in the nonregularized model; but under regularization, the testing error (which we are generally more interested in) is significantly lower. This is likely due to the smoothing effect presented in the textbook; since the estimate from the natural spline restriction is much less noisy, we are given a much better approximation of the testing error (although this constraint also increases our training error). Although both errors are somewhat high, this is likely due to imprecision in the creation of log-periodograms (and corresponding data values) of phonemes. Overall, it appears that the performance of logistic regression is superior to that of quadratic discriminant analysis, whether regularized or raw. Given the nature of the data, we should also note that QDA does not take into account any interaction or (auto)correlative terms. The data are quantifications of sound patterns over time, and as such there is likely a strong relationship between the data points. Although regularization helps, in order to reduce this error we would need some kind of model which takes into account serial correlation. Some type of data reduction as the first step (or identifying more clearly the inter-regressor correlations/relationships and accounting for this accordingly in the model) would probably be a better way to proceed in this problem, whether with respect to QDA, logistic regression, or any other method.