Question 1:

What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

The best alpha values for both regression models as found in the excercise are:

- Ridge: alpha = 7.0
- Lasso: alpha = 100

If we double the values of these alphas, we'll get:

- Ridge: alpha_doubled = 14.0
- Lasso: alpha_doubled = 200

After rebuilding the model using these alpha values:

- R-Squared for Ridge (Train) = 0.94
- R-Squared for Ridge (Test) = 0.90
- R-Squared for Lasso (Train) = 0.93
- R-Squared for Lasso (Test) = 0.89

The values of beta coefficients seem to be decreasing in case of Lasso if we double the value of alpha. This is because the model penalises more with increase in the value of alpha.

The most significant predictor variables after the change are:

Lasso:

OverallQual_Excellent	48308.622249
OverallQual_Very_Good	32186.695188
GrLivArea	23305.597375
Neighborhood_Crawfor	18302.452243
Functional_Typ	14758.305273

Ridge:

OverallQual_Excellent	25986.511816
OverallQual_Very_Good	22014.891918
GrLivArea	20564.601186
Neighborhood_Crawfor	13045.749873
OverallCond_Excellent	12131.473132

Question 2:

You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

As seen earlier, the difference between r2 scores for test and train datasets seem to be lower in case of Lasso. Since we have a large number of variables, feature selection will be handled by Lasso regression. So, I'd choose Lasso regression in this case.

Question 3:

After building the model, you realised that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

After deleting top 5 predictor variables from the dataset, the r2 values of the models was:

- R-Squared for Ridge (Train) = 0.93
- R-Squared for Ridge (Test) = 0.89
- R-Squared for Lasso (Train) = 0.94
- R-Squared for Lasso (Test) = 0.88

The r2 value for Ridge seems to have dropped slightly to 0.93 and 0.89. But the difference is higher in case of Lasso where for train data it is 0.94 and for test, it is 0.88. This means that for lasso, the model performance has dropped slightly when making predictions on test data.

The top 5 parameters for these two models now are:

Ridge without top 5:

2ndFlrSF	17883.235635
Neighborhood_NridgHt	15350.816201
1stFlrSF	15307.997827
Neighborhood_StoneBr	15086.860607
Neighborhood_NoRidge	14390.931962

Lasso without top 5:

Condition2_PosA	74001.308618
2ndFlrSF	20383.901375
Exterior1st_BrkFace	20366.731882
Neighborhood_StoneBr	19020.046501
Neighborhood_Somerst	17522.815604

Question 4:

How can you make sure that a model is robust and generalisable? What are the implications of the same for the accuracy of the model and why?

A model is said to be robust if any variation in the data doesn't drastically affect its performance.

A model is generalisable if it can adapt well to new, previously unseen data.

To make sure a model is robust and generalisable, we have to make sure the model doesn't overfit. A model is said to overfit if it performs very well on traning data with high accuracy, but fails miserably on unseen test data.
When a model is overfit, a slight variation in training data results in drastic decrease in the model performance in predicting target variable on test data.

To make sure our model doesn't overfit, it has to be not too complex. Simple models tend to have low variance and high bias, which means the model doesn't change much when there are any changes to training data. Since simpler models require less traning data, it is unlikely to overfit on the training data. Simpler models might have lower accuray on training data, but it performs well on new, unseen test data as it doesn't memorise the patterns in the training data set.

However, a very simple model can have very poor accuracy. So, we have to compromise on a point where the bias is not very high and the variance is also kept low to make a model robust and generalisable.