Report Q1:
According to Fornefett (paragraph 2.3, solvability), the polynomial part is not needed. This is due to usage of the Gaussian RBF, which is positive definite, hence can be calculated without the polynomial part.

Report Q2:
For the spline fitting problem I used the following formula:

$$\begin{pmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{q}_k \\ 0 \end{pmatrix}, \qquad (3)$$

Later W and λ are introduced:

$$\begin{pmatrix} \mathbf{K}+\lambda\mathbf{W}^{-1} & \mathbf{P} \\ \mathbf{P}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{q}_k \\ 0 \end{pmatrix}$$

The key formulas are in paragraph 2.1 under formulas 3 and 4.
The system needs to be solved with respect to alpha. The beta coefficient does not appear because according to the equation 2 it is the coefficient for the polynomial part.
The solution first involves formation of the K matrix:
$K_{ij} = R(\|\mathbf{p}_i - \mathbf{p}_j\|)$. This is done in the for loop:

```
for ii = 1:src_rown
    for jj = (ii+1):src_rown
        r(ii,jj)=norm(source_pts(ii,:)-source_pts(jj,:));
        r(jj,ii)=r(ii,jj); % size n x n
    end
end
```

All of that forms a linear system of type Ax = B, which needs to be solved with respect to x. Using the method described in https://uk.mathworks.com/help/matlab/ref/mldivide.html the linear system is solved with respect to alpha:

```
alpha = (K + lambda * W.^(-1)) \ target_pts;
```

Report Q3:
I used the mldivide method from Matlab, which is a least-squared solution of the system. I am aware about the singular value decomposition method that can be used here, but from my understanding it involves additional calculations - finding eigenvalues and eigenvectors, therefore I just used mldivide.

Report Q4:
From my understanding, the control points here are the points that can be set by a user and used to transform the image - the image will be translated/rotated/warped with respect to these points. I can choose to define any control points on the image, therefore from my understanding any points can be chosen.

Report Q5:
Lambda is not needed at the evaluation stage.

Report Q6:
I used built-in Matlab functions that operate on the whole array of the pre-computed Euclidean distance set. Probably, further Euclidean distance calculation of query and control points could be vectorised using meshgrid for 1:qry_rown and 1:control_pts and applying norm on the whole array at once.

Report Q7:
Sigmas are the individual weights of the landmarks, representing landmark localisation errors.
Report Q8-Q12:
Unfortunately I don’t have the knowledge and not competent to answer these questions. Despite the fact I have worked for the month at least 8 hours a day everyday on that coursework, I am unable to complete it.