1. Consider the 8-level midrise uniform quantizer covered in the lecture. The quantization boundaries are at $0, \pm\Delta$, $\pm 2\Delta$.., and the quantization levels are at $\pm\frac{1}{2}\Delta$, $\pm\frac{3}{2}\Delta$...For an input of value 3.8Δ, after quantization, the result will be **1 point**

- ○ $\frac{9}{2}\Delta$
- ○ $-\frac{3}{2}\Delta$
- ○ $-\frac{5}{2}\Delta$
- ◉ $\frac{7}{2}\Delta$

2. (Yes or No) Can we perform non-uniform quantization using a uniform quantizer? **1 point**

- ◉ Yes
- ○ No

3. Which of the following statements regarding Vector Quantization (VQ) are correct? Check all that apply. **1 point**

- ☑ VQ can be considered as a pattern matching technique.
- ☐ The codevectors in VQ are of lower-dimensionality than the input vectors.
- ☐ The designed codebook is unique irrespectively of the training vectors.
- ☑ Codebook design is an iterative procedure.

4. For a given image of fixed resolution, how is the number of bits resulted from Vector Quantization (VQ) related to the vector size and the number of codevectors? **1 point**

- ○ Larger vector size leads to fewer bits; more codevectors lead to fewer bits
- ◉ Larger vector size leads to fewer bits; fewer codevectors lead to fewer bits
- ○ Smaller vector size leads to fewer bits; more codevectors lead to fewer bits
- ○ Smaller vector size leads to fewer bits; fewer codevectors lead to fewer bits

5. Which of the following options are isometries? Check all that apply **1 point**

- ☑ identity
- ☑ reflection about the mid-vertical axis
- ☑ flipping about the diagonal
- ☑ the concatenation of rotation by 90 degrees with rotation by -90 degrees

6. In a lossy image compression technique such as JPEG, from which of the following does the loss come from? **1 point**

- ○ Dividing an image into blocks
- ○ Transformation of image blocks
- ◉ Quantization of transformation coefficients
- ○ Run-length coding of quantized coefficients

7. In this problem you will get hands-on experience in JPEG image compression. Follow the instructions below to complete this problem. **1 point**

(1) Download the original 8-bit grayscale image here, and load it into a MATLAB array.

(2) Perform JPEG compression by using the MATLAB function "imwrite". For the purpose of this problem, you need to specify 5 input arguments. The first argument is the MATLAB array containing the input image; the second argument is a string specifying the output file name; the third argument is 'jpg' (including the single quotes); the fourth argument is the string 'quality' (including the single quotes); and the last argument is a number that specifies the quality level used for compression. The quality level is an integer between 0 and 100. For this step, set the quality level to 10. After the function "imwrite" is invoked, a new JPEG image will be created in the location that was specified by you.

(3) Load the newly created JPEG image into a MATLAB array. Compute the PSNR between the JPEG compressed image and the original image. Note that the image loaded into MATLAB is of type 'uint8' (i.e., 8-bit integer). In order to compute the PSNR, you need to convert these arrays into 'double'.

(4) Enter the PSNR value (up to 2 decimal points) in the box below.

26.43