1. We can represent the convolution operation when the kernel is centered on the pixel of interest by? Note that I and K represents the image and the kernel respectively.

   A. $S_{ij} = \sum_{a=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a,j-b} K_{a,b}$

   B. $S_{ij} = \sum_{a=\lfloor -\frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b=\lfloor -\frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a,j-b} K_{\frac{m}{2}+a,\frac{n}{2}+b}$

   C. $S_{ij} = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} I_{i+a,j+b} K_{a,b}$

   Solution: Option B is the correct answer.

2. What is the output dimension of the resulting image when a 7×7 kernel is applied to a 9×9 image?

   A. 3×3

   B. 5×5

   C. 2×2

   Solution: Option A is the correct option.

3. Given a $128 \times 128 \times 3$ image and 6 filters of size $9 \times 9 \times 3$, what will be the dimension of the output volume when a stride of 1 and a padding of 2 is considered?

   A. $124 \times 124 \times 6$

   B. $119 \times 119 \times 6$

   C. $121 \times 121 \times 6$

   Solution: Using the formula $W_2 = \frac{W_1 - F + 2P}{S} + 1$ and $H_2 = \frac{H_1 - F + 2P}{S} + 1$, we get $W_2 = H_2 = 128 - 9 + 4 + 1 = 124$ and $D_2 = 6 =$ no. of filters. Hence, **Option A** is the correct answer.

4. Consider the following Convolutional neural network where all the convolution filters are of size 3×3. For all the convolution layers, the stride $S = 1$ and padding $P = 1$:

   - **CONV1**: convolutional layer which takes an image of size $28 \times 28 \times 1$ as input, and produces 64 outputs (64 filters of size $3 \times 3 \times 1$)
   - **POOL1**: 2×2 max-pooling layer

- **CONV2**: convolutional layer with 64 inputs, 128 outputs (128 filters of size 3 $\times$ 3 $\times$ 64)
- **POOL2**: 2 $\times$ 2 max-pooling layer
- **CONV3**: convolutional layer with 128 inputs, 256 outputs
- **CONV4**: convolutional layer with 256 inputs, 256 outputs
- **POOL3**: 2 $\times$ 2 max-pooling layer
- **FC1**: fully connected layer with 1024 outputs

What is the number of parameters in the FC1 layer?

A. 256 * 1024

B. 4096 * 1024

C. 1024 * 1024

Solution: The number of parameters in the model are :

Layer	**Input Size**	**Output Size**	**# Parameters**
Conv 1	(28,28,1)	(28,28,64)	$3 \times 3 \times 64 = 576$
Max Pool 1	(28,28,64)	(14,14,64)	0
Conv 2	(14,14,64)	(14,14,128)	$3 \times 3 \times 128 \times 64 = 73728$
Max Pool 2	(14,14,128)	(7,7,128)	0
Conv 3	(7,7,128)	(7,7,256)	$3 \times 3 \times 128 \times 256 = 294912$
Conv 4	(7,7,256)	(7,7,256)	$3 \times 3 \times 256 \times 256 = 589824$
Max Pool 3	(7,7,256)	(4,4,256)	0
FC 1	(4096,1)	(1024,1)	$4096 \times 1024 = 4194304$
FC 2	(1024,1)	(1024,1)	$1024 \times 1024 = 1048576$
Output	(1024,1)	(1024,10)	$1024 \times 10 = 10240$

Therefore, the parameters of FC1 would be 4096 * 1024. Hence, **Option B** is the correct answer.

5. Consider the problem where we have a Feed-forward Neural Network (FNN) with 2 layers such that it takes a m-dimensional vector as input and produces an a n-dimensional vector as output. Is it possible to represent this FNN as a Convolutional Neural Network (CNN)?

A. No, it is not possible.

B. Yes, it is possible, by choosing m filters of size $n \times 1$.

C. Yes, it is possible, by choosing n filters of size $m \times 1$.

Solution: Option C is the correct answer.
We choose **n** filters, each of size **(m x 1)**, with stride as **1**. We also do not pad the input vector with zeroes as it is not required in this case. The input vector is convolved with each of the **n** filters and we will get **n** scalars. We can then stack up the n scalars into a **(n x 1)** vector which is the desired output. The above CNN model actually represents a FNN where we perform a matrix vector multiplication of the input vector of size **(1 x m)** with a weight matrix of size **(m x n)** to get a **(1 x n)** output vector.

6. Is the max-pooling layer differenciable?

    A. True

    B. False

Solution: Option A is the correct answer

7. In the context of Deep Art, if $V \in \mathcal{R}^{32\times(128\times128)}$ is the activation at a layer, then which of the following captures the style of the image?

    A. $V^\top$

    B. $V^\top V$

    C. $V^{-1}V$

Solution: Option B is the correct answer.

8. Consider an image $\in \mathbb{R}^{28\times28}$ which is stored as a numpy array of size 28×28. You can download the numpy array from here :

    https://drive.google.com/drive/folders/1JRcAijvg2cngfamBF5tv9m33RqKoxo8D?usp=sharing

    What will be the result of applying the following kernel to this image ?

$$\begin{bmatrix} 0.053 & 0.110 & 0.054 \\ 0.111 & 0.225 & 0.111 \\ 0.054 & 0.110 & 0.053 \end{bmatrix}$$

    A. it will result in a sharpened image

B. it will result in a blurred image

C. it will result in an image which contains all the edges in the original image

D. it will result in a black and white version of the original image

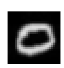
Solution: On plotting the image contained in the .npy file, the original image looks like the following :

You can apply the give filter onto the above image using the following code:

```
import cv2
import numpy as np

# Load the image
canvas_load = np.loadtxt("Q8_image.npy")
# To visualise and store the original image
cv2.imwrite("original.png", canvas_load)
# Define the kernel/filter
kernel = np.array([[0.053, 0.110, 0.054],
                   [0.111, 0.225, 0.111],
                   [0.054, 0.110, 0.053]])
# Apply the filter onto the image
dst = cv2.filter2D(canvas_load, -1, kernel)
#Save the filtered image
cv2.imwrite("./filtered.png", dst)
```

The filtered image looks like the following:



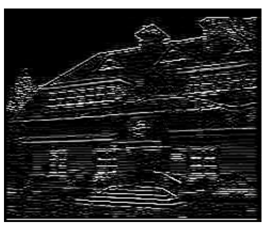
You can clearly see that the filtered image is a blurred version of the original image. Hence, **Option B** is the correct answer.

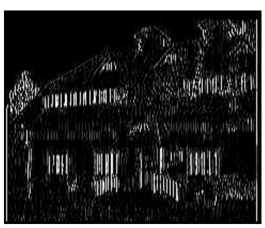
9. Consider the image shown below:

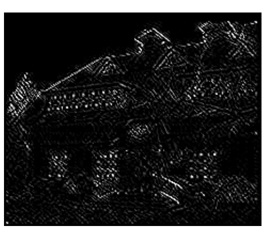
Figure 1: House Image

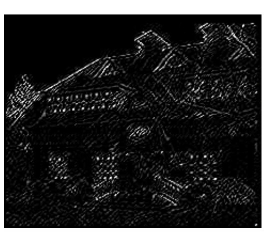
What will be the result of applying the following kernel to the above image ?

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

A. 

B. 

C. 

D. 

Solution: The given kernel acts as a horizontal edge detector i.e on applying the kernel onto the house image, it will detect the horizontal edges. You can also apply verify the above by executing a similar code as mentioned in the previous question. Hence, **Option A** is the correct option.