1. What is TensorFlow? Which company is the leading contributor to TensorFlow?
    a. TensorFlow is an open-source software library used for Neural Network applications. Google is the leading contributor to TensorFlow
2. What is TensorRT? How is it different from TensorFlow?
    a. TensorRT is a platform for high-performance deep learning inference. TensorRT focuses on inference (not training) but when integrated with Tensorflow, it allows for faster inference.
3. What is ImageNet? How many images does it contain? How many classes?
    a. ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. It has over 14 million images, with more than 21,000 classes.
4. Please research and explain the differences between MobileNet and GoogleNet (Inception) architectures.
    a. "GoogLeNet," was the 22-layer winner of the ILSVRC 2014 competition. Also known as Inception, the idea was to use parallel convolutions between layers and let the model decide which to use. Inception demonstrated the power of well-designed "network-in-network" architectures, adding yet another step to the representational power of neural networks. In order to solve the computational bottleneck, the authors of Inception used 1x1 convolutions to "filter" the depth of the outputs. A 1x1 convolution only looks at one value at a time, but across multiple channels, it can extract spatial information and compress it down to a lower dimension.
    b. MobileNet is essentially a streamlined version of the Xception architecture optimized for mobile applications. The hypothesis of Xception is that cross-channel correlations and spatial correlations are sufficiently decoupled that it is preferable not to map them jointly. In a traditional conv net, convolutional layers seek out correlations across both space and depth. Rather than partitioning input data into several compressed chunks, Xception maps the spatial correlations for each output channel separately and then performs a 1x1 depthwise convolution to capture cross-channel correlation.
    c. Xception outperforms GoogLeNet on ImageNet dataset and has the same number of model parameters
5. In your own words, what is a bottleneck?

    a. Answer: A bottleneck is a term we use to describe the layer that actually does the classification, which is found just before the final output layer. We use the term bottleneck because near the output, the representation is much more compact than in the main body of the network.
6. How is a bottleneck different from the concept of layer freezing?
    a. Answer: The bottleneck is the final layer in the network. With layer freezing, we freeze the network up to this last layer. This way, we can reuse a pre-trained model's weights, but we allow the final layer to be trained using our training data. This is the concept of transfer learning.

7. In the TF1 lab, you trained the last layer (all the previous layers retain their already-trained state). Explain how the lab used the previous layers (where did they come from? how were they used in the process?)
    a. Answer: TF Hub distributes models without the top classification layer (headless models). We created a feature extractor and froze the variables in the feature extractor layer, so that the training only modifies the final layer. Finally, we added a new classification layer at the end which was our classification head.

8. How does a low --learning_rate (step 7 of TF1) value (like 0.005) affect the precision? How much longer does training take?
    a. Training took only 0.5 seconds longer. Training accuracy dropped from .91 to .88 when learning rate was .005.

9. How about a --learning_rate (step 7 of TF1) of 1.0? Is the precision still good enough to produce a usable graph?
    a. The precision is not good enough to produce a usable graph. The Validation accuracy dropped down to 0.82

10. For step 8, you can use any images you like. Pictures of food, people, or animals work well. You can even use ImageNet images. How accurate was your model? Were you able to train it using a few images, or did you need a lot?
    a. I added the category "Avocado" to my flower images. My model was not as accurate, because I only used 30 images of avocados, so I ended up with about 0.84 accuracy. However, I did add some test images of avocados and my retrained model was able to predict them as avocados with 99%

accuracy:
python3 -m scripts.label_image --graph=tf_files/retrained_graph.pb --image=val_avocado/pexels-photo-142890.jpeg

b. The validation images can be found in val_avocado

11. Run the TF1 script on the CPU (see instructions above) How does the training time compare to the default network training (section 4)? Why?
    a. Training time is slower compared to GPU (> 2 minutes). This is because GPUs are better optimized for training neural networks because of their high bandwidth. They have higher bandwidth memory and larger and faster register L1 memory which is easily programmable. GPUs enable more storing of data in their L1 caches and register files to reuse convolutions and matrix multiplications.

12. Try the training again, but this time do export ARCHITECTURE="inception_v3" Are CPU and GPU training times different?
    a. GPU training time: 2m 33s
    b. CPU training time: 3m 30s
13. Given the hints under the notes section, if we trained Inception_v3, what do we need to pass to replace ??? below to the label_image script? Can we also glean the answer from examining TensorBoard?
    a. `python -m scripts.label_image --input_layer= 'Mul:0' --input_height=299 --input_width=299 --graph=tf_files/retrained_graph.pb --image=tf_files/flower_photos/daisy/21652746_cc379e0eea_m.jpg`