1) Movie reviews and the model's predictions on them:-

   a) Review - Christopher Nolan's Inception is an awesome film
      Prediction - 0.9860913157463074

   b) Review - The film I watched yesterday is a horrible one
      Prediction - 0.023268666118383408

2) Answers to the inline questions -

a) Why is the residual connection is crucial in the Transformer architecture?

Residual connections are normally used to mitigate the problem of vanishing gradients, especially in very deep neural networks. At the time of backpropagation, the gradient gets multiplied by the derivative of the activation function and for most of the activation functions, this product becomes very less or close to zero. Consequently, a large part of the training signal would get lost during back-propagation. The summation operations in case of Residual connections are always linear with respect to the derivative of the activation functions, hence, a part of the previous gradient will always flow through this part and not get affected by the vanishing gradient. In the case of Transformers, in addition to the above, these connections help to keep the information local in the Transformer layer stack. The self-attention mechanism allows an arbitrary information flow in the network and thus arbitrary permuting the input tokens. To some extent, the residual connections give a guarantee that contextual representations of the input tokens really represent the tokens by always telling the Transformer the representation of what the original state was.

b) Why is Layer Normalization important in the Transformer architecture?

Layer Normalization helps us to speed up the training process without having any dependence on the batch size during training as it does the normalization task along the feature dimension and not the batch dimension. This makes it very helpful to use in RNNs where the normalization statistics are very different across batches and batch normalization fails to work properly. Moreover, Layer Norm can work with varying input size unlike Batch Norm, which helps RNNs and self-attention based models like Transformers that can have inputs of varying size. In addition to this, the Transformer architecture was designed for RNN/NLP tasks mainly.

c) Why do we use the scaling factor of 1/d_k in Scaled Dot Product Attention? If we remove it, what is going to happen?

   The value of the dot product becomes large when the value of d_k is large. Consequently, the softmax values in this case will go into regions where it will produce very small gradients and the model's weights will not get updated properly. That is why we divide the dot product value by sqrt(d_k) to prevent the inputs to the softmax function from becoming very large. If we remove it, for small values of d_k, it might not create any problem, but for large values, the inputs to the softmax function will become large and the gradients will become very small.