Assignment 3

(Q1) The question asks us to explain why Decoder-only Transformer is more scalable than Encoder-Decoder architecture for LLMs. Firstly, I would like to clearly show that the Decoder-only is enough for our task and then tell why they are more Scalable.

Since we are talking about LLMs, lets make it clear and agree that they are trained to do only one thing which is their main task ⇒ predict the next token given previous other tokens as input. This task as we know is autoregressive and a decoder-only model does this using 'masked self-attention' ensuring that it only sees past tokens and never see the future (tokens). Adding an encoder is not needed since there is no separate input sequence that needs to be encoded and attended to. Cross-attention mechanism is not needed and everything can be handled by self-attention only.

Decoder only models are more scalable due to multiple reasons, firstly

training simplicity → since there is no alignment between input/output sequence.
→ works on any text (so no problem of deficit training data)

inference efficiency → caching key-value attention states increases efficiency.
→ is fast which is crucial for long conversations

unification of tasks → just by prompting, it can do any task with any need for architectural changes.

Q2) It is asked why is 'Next - Token Prediction' sufficient to learn complex language abilities like reasoning, translation and summarization. So lets talk about translation, reasoning and summarization. These are not different / totally ~~separate~~ separate objectives. These are just different ways / structures of writing / how humans write. So, when working with LLMs, the full probability distribution of language is modelled and the LLMs are forced to internally learn the different structures / patterns that can help in text generation. Translation involves learning the map between the two languages. Reasoning involves understanding the logic / rule for right prediction. Summarization involves understanding the context and returning a compressed form of the inputs. We need not ~~[illegible]~~ ~~have~~ teach it everything separately, instead giving a huge dataset involving all these tasks already during training makes it learn the different ways which results in altering the prediction values and hence the Next Token Prediction is sufficient for all these tasks. ~~[illegible]~~ Because after probability distribution is known, only choosing or predicting the next token is left.