# START OF QUIZ
# Student ID: 29404548,Lee, Ai Ka

## Question 1

Topic: Lecture 3
Source: Lecture 3

What are the benefits of using adapter layers instead of fine-tuning? (1)

## Question 2

Topic: Lecture 2
Source: Lecture 2

Describe the concept of negative transfer with an example. (1)

## Question 3

Topic: Lecture 4
Source: Lecture 4

What are the differences between hard and soft EM, and why do they matter? (1)

## Question 4

Topic: Lecture 1
Source: Lecture 1

Many languages lack a standardized writing system. How does this impact the creation of NLP tools, and how might we approach building CL tools for such languages? (1)

## Question 5

Topic: Lecture 4
Source: Lecture 4

Are there any situations where the alpha and beta score at a particular timestep would be equal? (1)

## Question 6

Topic: Lecture 3
Source: Lecture 3

Imagine that we find a database lying around, and it's been very poorly maintained and documented. All we know is that it contains word embeddings for a language written in Arabic script (assume we can't read it, and the only Arabic speakers we know also can't read it - it's in a language they don't know). How might we go about trying to identify the language that it's written in, without finding speakers of all of the Arabic-derived languages? (2)

## Question 7

Topic: Lecture 2
Source: Lecture 2

In transfer learning, how do you decide which layers of a pre-trained model to freeze and which to fine-tune when adapting it to a new language or task? Give an example of when you might choose to freeze or fine-tune specific layers. (2)

## Question 8

Topic: Lecture 1
Source: Lecture 1

Many existing tools and annotation formats make assumptions about the languages that they are processing. If you were creating an ML corpus for a new language, would you prefer to start from scratch, or to adapt an existing annotation schema? Would this change depending on if you were working with a Class 1 or a Class 5 language? Explain. (2)

## Question 9

Topic: Long
Source: Lecture 3

Imagine that we want to take what we know about adapter layers and word embeddings to approach shared embedding space in a very different way. We have several multi-lingual embeddings in HRLs that we know are in the same space. We also have embeddings that we've trained for a LRL, but that are in a different space. We concatenate the embeddings, freezing the HRLs, but not the LRL embeddings, and then pass them through a prediction layer for POS tagging. Do you think this would work? Would it be better to try to predict the HRL or LRL (or do it as multi-task learning)? (3)

# END OF QUIZ