Artificial Intelligence Final Report Assignment 問題3 (Problem 3)

レポート解答用紙 (Report Answer Sheet)

Group Leader

学生証番号 (Student ID): 21522804

名前(Name): Phạm Hoài Vũ

Group Members

学生証番号 (Student ID): 21522885

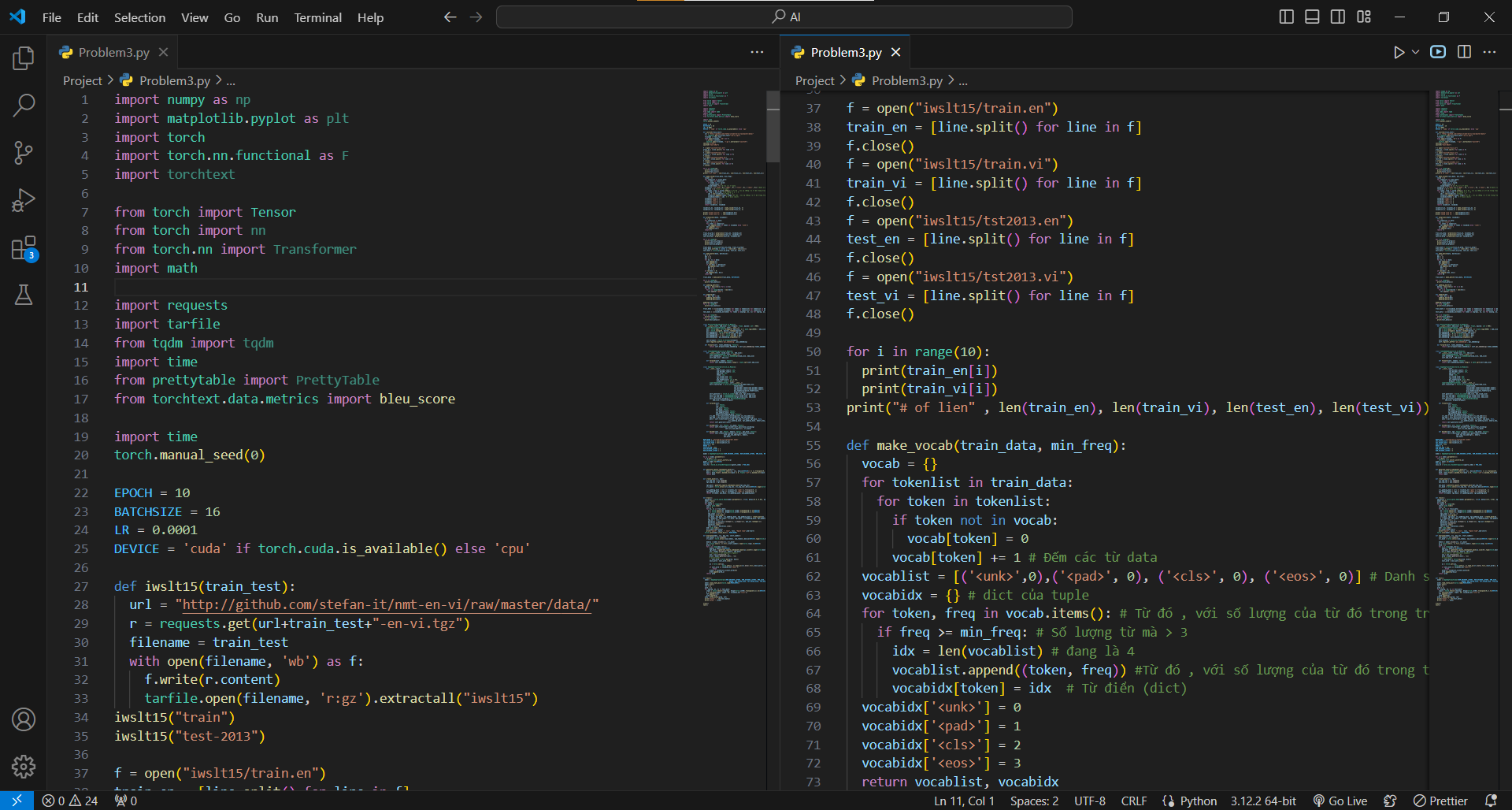
名前(Name): Phan Thị Cát Tường

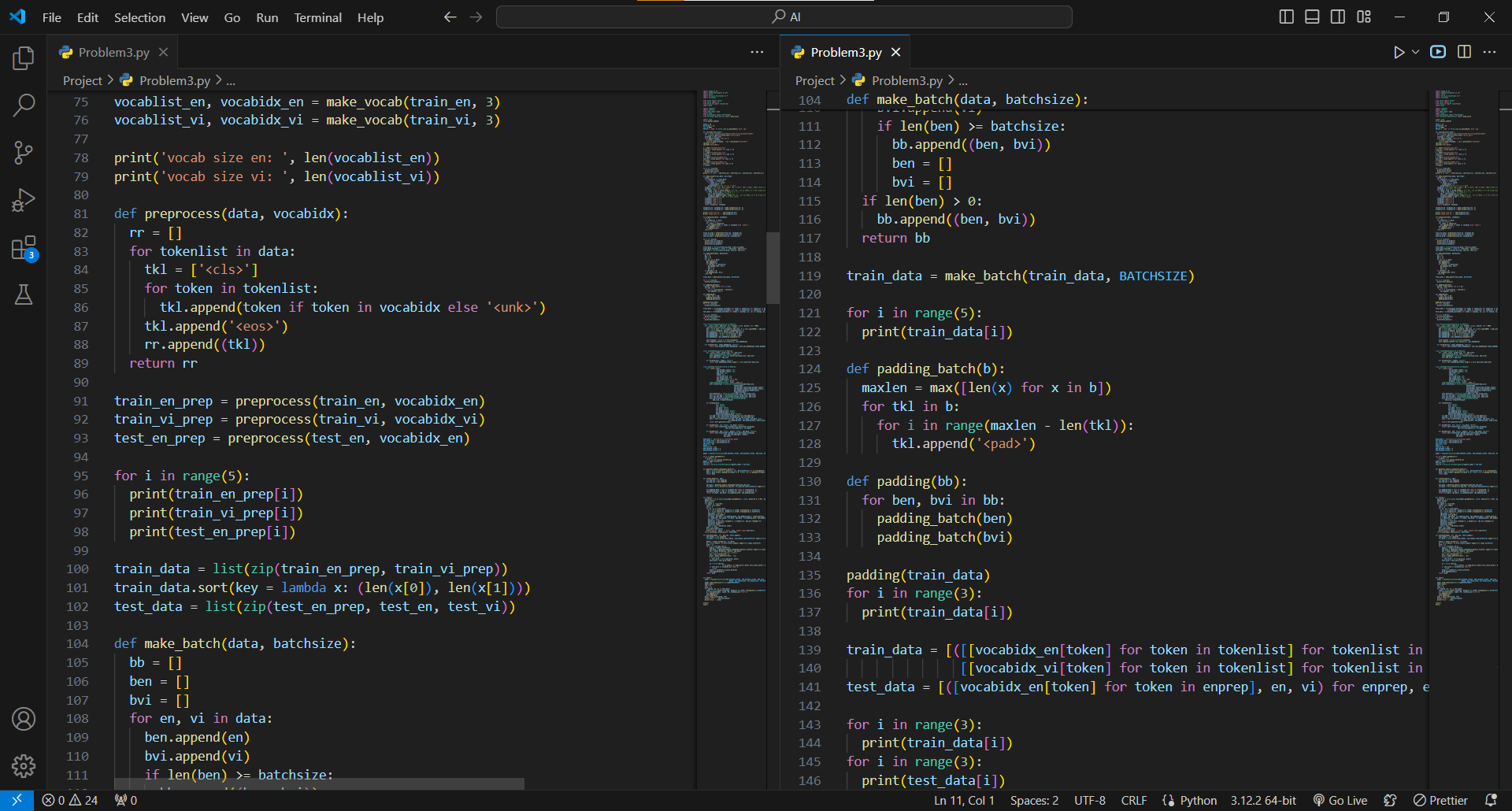
学生証番号 (Student ID): 21522747

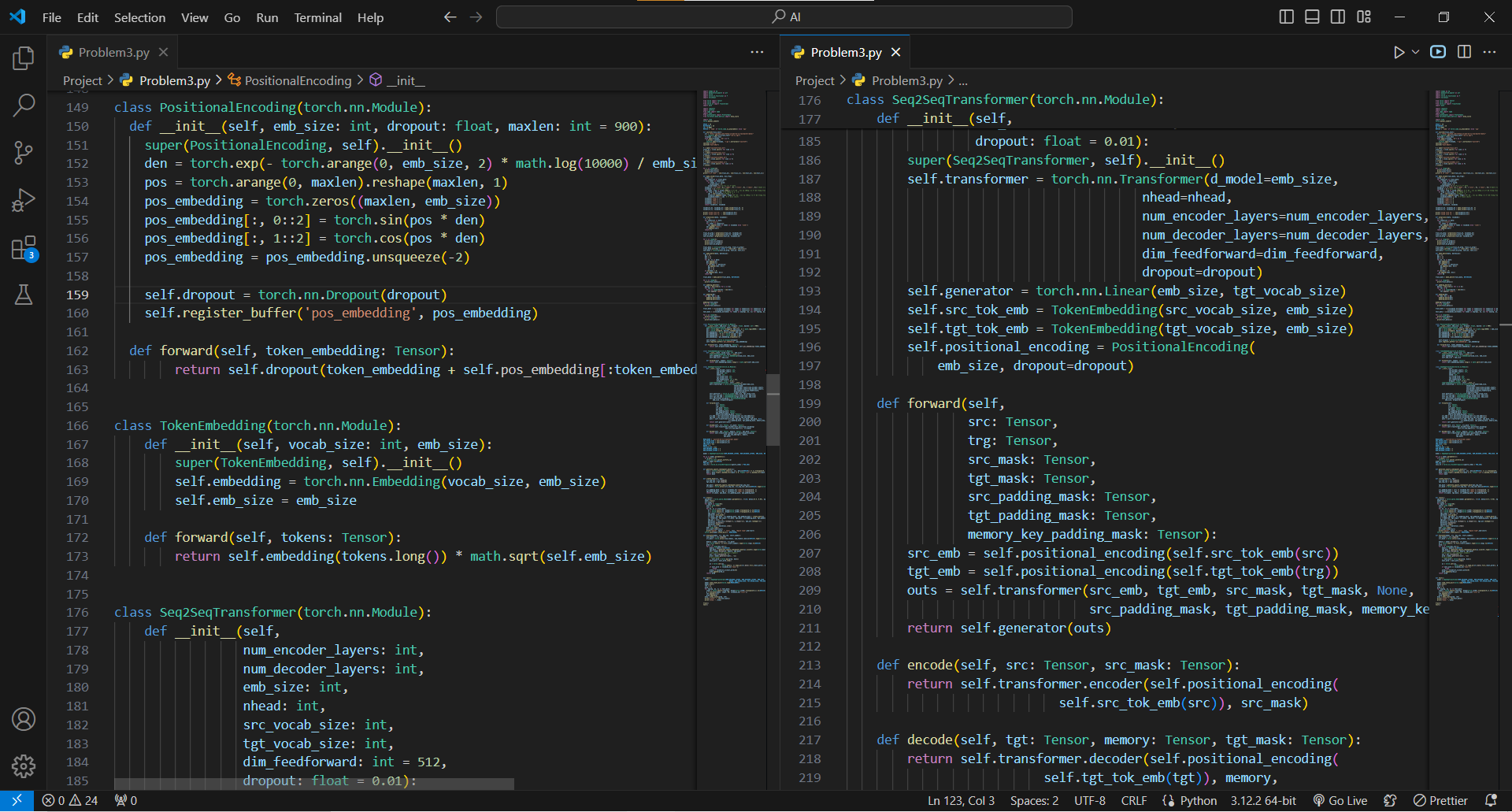
名前(Name): Trịnh Tuấn Tú

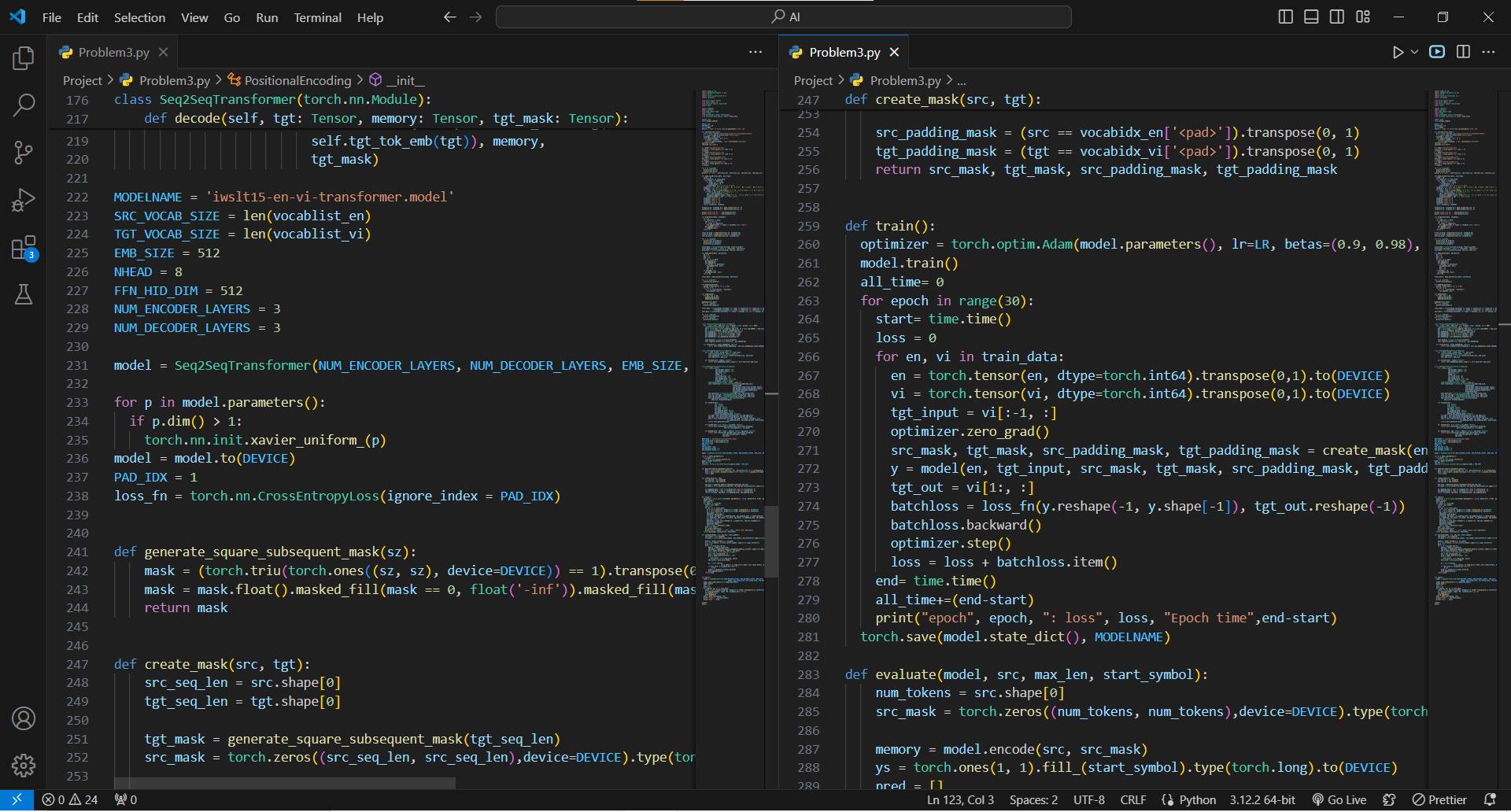
問題3 (Problem 3)のレポート

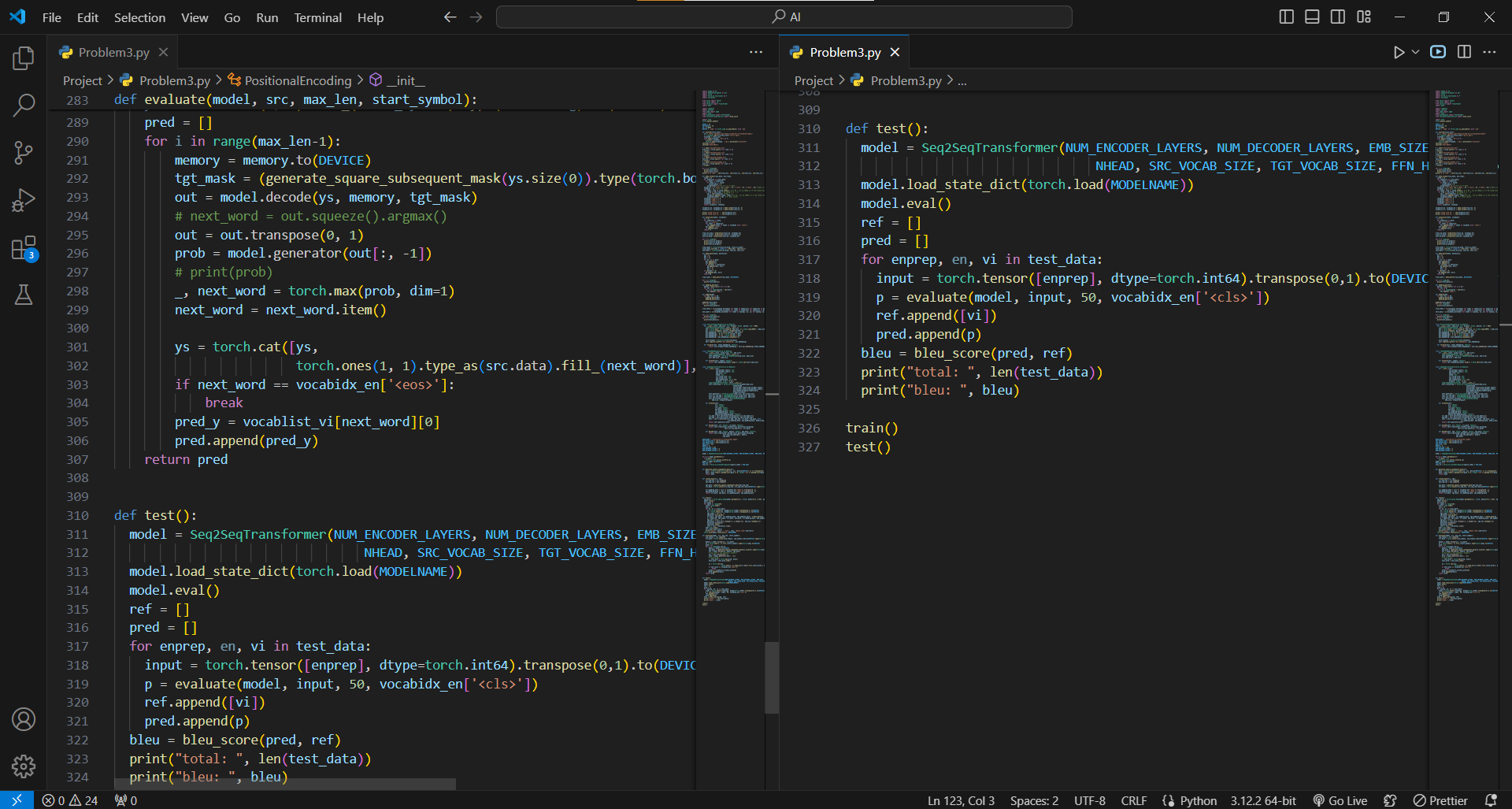
I. Program:





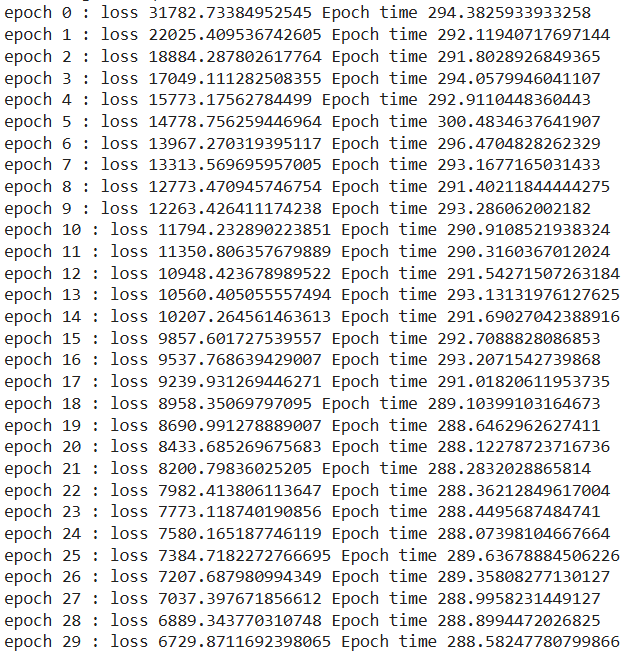






II. Execution Results

* **Loss in each epoch:**



* **Bleu Score in Test dataset:**

****

**III. Explanation**

* In the improved program, several changes and improvements were made to increase accuracy. Here are the key changes:
  + Data:
    - The make\_vocab function is used to build a dictionary from training data, removing low-frequency words and ensuring special tokens always have fixed indices.
    - The preprocess function adds special tokens (<cls>, <eos>) and replaces out-of-vocabulary words with <unk>.
    - Create train\_data and test\_data lists by combining pre-processed sentences.
    - The make\_batch function divides data into batches of size batchsize.
    - padding\_batch and padding functions add <pad> tokens to ensure all sentences in a batch have the same length.
    - Token-to-index conversion: Converts tokens in batches into indices based on the dictionary.
  + Model: The model has been modified compared to the programs in the 13th lectures. We use a Transformer model. This model can process sequential inputs without relying on mechanisms like LSTM or GRU used in previous models. Instead, it employs a mechanism called "self-attention" to understand relationships between words within a sentence.
  + Model Architecture:
    - Key components of the model:
      * PositionalEncoding: Adds positional information of words in sentences to embeddings.
      * TokenEmbedding: Generates embeddings for tokens from the dictionary.
      * Seq2SeqTransformer: Constructs a Transformer model for the Seq2Seq task.
    - Supporting functions:
      * generate\_square\_subsequent\_mask: Creates masks for the decoding process to only allow each word to see preceding words.
      * create\_mask: Generates masks for the encoding and decoding processes.
    - Model parameters:
      * NUM\_ENCODER\_LAYERS: Number of encoder layers in the Transformer model.
      * NUM\_DECODER\_LAYERS: Number of decoder layers in the Transformer model.
      * EMB\_SIZE: Size of the embedding vector.
      * NHEAD: Number of heads in multihead attention.
      * FFN\_HID\_DIM: Size of the hidden layer in the feed-forward network.
      * SRC\_VOCAB\_SIZE: Vocabulary size of the source language.
      * TGT\_VOCAB\_SIZE: Vocabulary size of the target language.
      * BATCH\_SIZE: Batch size during training.
      * NUM\_EPOCHS: Number of epochs for training.
      * DEVICE: Device used for training the model (CPU or GPU).
    - Training parameters:
      * BATCH\_SIZE: Batch size during training.
      * NUM\_EPOCHS: Number of epochs for training.
      * DEVICE: Device used for training the model (CPU or GPU).
      * optimizer: Optimization algorithm (such as Adam optimizer) used to update model weights.
      * loss\_fn: Loss function (such as CrossEntropyLoss) used to evaluate model performance.
  + Training and evaluation:
    - train function: Trains the model, computes loss, and performs optimization steps.
    - evaluate function: Evaluates the model, computes loss on the validation set.