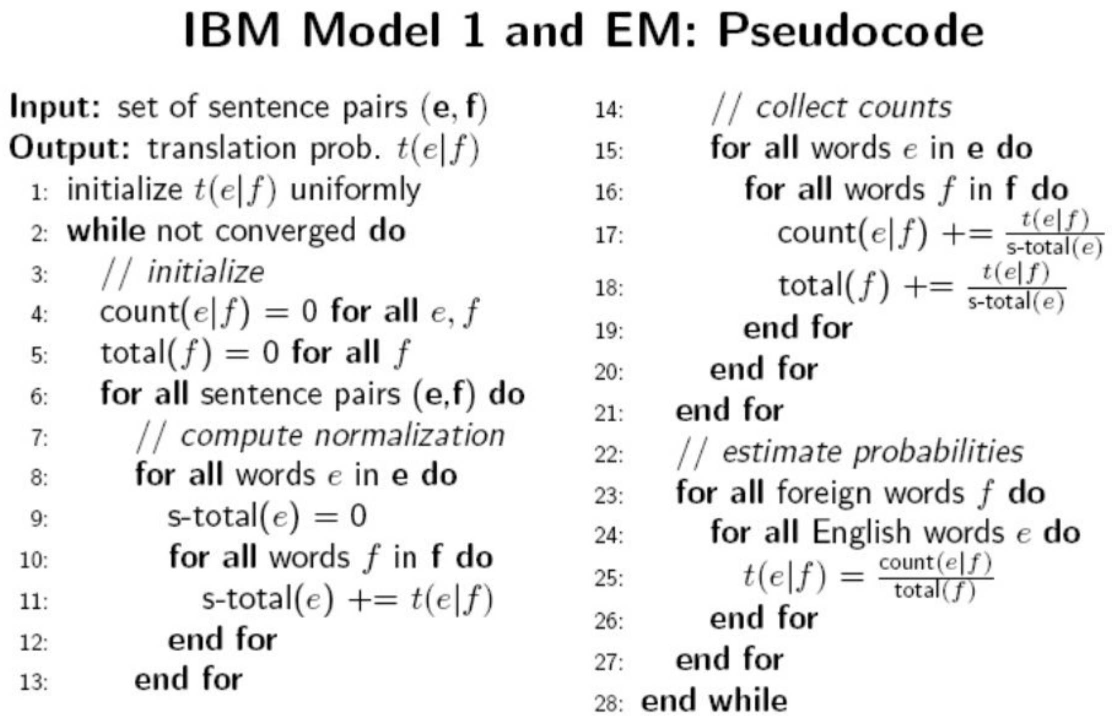
Hint:

(1) The pseudocode below might help with your implementation:



IBM Model 1 and EM: Pseudocode

```
Input: set of sentence pairs (e, f)
Output: translation prob. $t(e|f)$
initialize $t(e|f)$ uniformly
while not converged do
  // initialize
  count($e|f$) = 0 for all $e, f$
  total($f$) = 0 for all $f$
  for all sentence pairs (e,f) do
    // compute normalization
    for all words $e$ in e do
      s-total($e$) = 0
      for all words $f$ in f do
        s-total($e$) += $t(e|f)$
      end for
    end for
    // collect counts
    for all words $e$ in e do
      for all words $f$ in f do
        count($e|f$) += $\frac{t(e|f)}{\text{s-total}(e)}$
        total($f$) += $\frac{t(e|f)}{\text{s-total}(e)}$
      end for
    end for
  end for
  // estimate probabilities
  for all foreign words $f$ do
    for all English words $e$ do
      $t(e|f) = \frac{\text{count}(e|f)}{\text{total}(f)}$
    end for
  end for
end while
```

2, Example:

Input:

s1_src = “the house”

s1_tgt = “ la maison”

s2_src = “ house”

s2_tgt = “maison”

Iteration_number = 2

Output:

t(la|the) = 0.625

t(maison|the) = 0.375

t(la|house) = 0.172

t(maison|house) = 0.828

3- Following Questions 1&2, please write a program to compute the lexical probabilities of any word pairs given a parallel corpus (train.en, train.de), where train.fr is the source data file and train.en is the target file. The output should be a file which contains word pairs with their translation probabilities.

Input: