Problem Statements

1. Given $k, L \in \mathbb{Z}$ with $1 \leq k \leq 5$ and $10 \leq L \leq 10000$, and

$$p \in \Delta^{4^k-1} = \left\{ p \in \mathbb{R}^{4^k} : p_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{4^k} p_i = 1 \right\}$$

a probability distribution over the set of all k-mers (ordered lexicographically). Design an algorithm that constructs a DNA sequence s with $\text{len}(s) = L$ such that

$$\frac{F_k(s)}{\|F_k(s)\|_1} = p.$$

In other words, the normalized k-mer composition of s should match the given probability vector p. Remember that the k-mer composition vector of a DNA sequence s, denoted

$$F_k(s) = [f_1, f_2, \ldots, f_{4^k}],$$

is defined such that f_i is the number of occurrences of the i-th k-mer in s.

Program Specifications:

- Your solution **must be implemented in Python** and saved in a file named `q_11.py`.
- The program should accept the following command-line arguments:
  - `--k`: a string representing the integer k. $1 \leq k \leq 5$.
  - `--L`: a string representing the integer L. $10 \leq L \leq 10000$
  - `--p`: the path to an NPZ file containing the probability vector $p \in \Delta^{4^k-1}$.
- The program will be executed as follows:

```
python q11.py --k 2 --L 100 --p test_n.npz
```

2. Similar to point mutations in biological sequences, sequencing errors frequently result in **single-nucleotide substitutions**. In a sequencing dataset, these errors manifest as reads that appear **only once** and differ by **exactly one nucleotide** from another, more frequent read (or its reverse complement). Correcting such errors is essential for improving the accuracy of downstream genomic analyses.

   Input: A collection of up to 1000 reads, each of **equal length** (maximum **50 bp**) in FASTA format. For each read s in the dataset, exactly one of the following holds:
   (a) **Correct Reads:** The read **appears at least twice** in the dataset (or as its reverse complement).
   (b) **Erroneous Reads:** The read appears exactly once but is one substitution away from a correct read in the set (or its reverse complement).

   Output:
   Return a list of all necessary corrections in the format:

   ```
   "[old read]->[new read]"
   ```

   where:
   - `old read` is the incorrect read.
   - `new read` is the corrected version.
   - Each correction must be **exactly one symbol substitution**.
   - The order of the corrections in the output does not matter.

   Implementation Details:

   - Your solution **must be implemented in Python** and saved in a file named `q13.py`.
   - Your program must accept the following argument: `--reads`: The name of the FASTA file containing the reads.
   - Your program will be executed as follows:

   ```
   python q13.py --reads test_n.fasta
   ```