1. In the CLOSEST STRING problem, we are given k strings $s^1, \dots, s^k$ of length n from an alphabet of size ℓ, and an integer d. The goal is to find a string s such that its Hamming distance from each of $s^1, \dots, s^k$ is at most d where Hamming distance between two strings is the number of indices where their letters do not match.

   We want to show that CLOSEST STRING is FPT with parameter k. We can view the input as a matrix with k rows and n columns.

   a) Show that we can bound $\ell \leq k$.

   b) Show that we can bound $n \leq k^k$, i.e., that the number of columns can be upper-bounded. This shows that there are at most k^k different *types* of columns on input.

   c) If we write the output string s under the matrix, then we can freely permute the columns of this $(k+1) \times n$ matrix and this will not affect the Hamming distance between s and $s^1, \dots, s^k$. So in a sense, it does not matter where a column of each type is, only to which output character it maps.

   d) Using this fact, write an integer program and conclude that you have a double exponential algorithm from Lenstra's algorithm.

   e) Observe that your integer program can be rewritten as an n-fold. What are the algorithmic consequences?

   f) Consider the following modification of CLOSEST STRING. Instead of having just a single integer d as input, we have integers $d_1, \dots, d_k$, and the goal is now to find a string s such that its Hamming distance from s^i will be at most d_i for every $i \in [k]$. Will our approach via integer programming still work?

2. In the PLANAR VERTEX COVER problem, we are looking for a vertex cover of size at most k in a graph G that is planar.

   Find an algorithm with running time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$.