1. Show that CLIQUE and INDEPENDENT SET admit an FPT algorithm on r-regular graphs where the parameter is the solution size k and r is a fixed constant. r being a constant means that $r \in \mathcal{O}(1)$, so e.g. $n^{\mathcal{O}}(r)$ is a fine running time.

   Then also prove that there is an FPT algorithm when r is also a parameter and no longer a constant.

2. In the CLUSTER VERTEX DELETION problem, we are given a graph G and an integer k, and the task is to delete at most k vertices from G to obtain a cluster graph (a disjoint union of cliques). Obtain a $3^k n^{\mathcal{O}(1)}$-time algorithm for Cluster Vertex Deletion.

3. Describe an algorithm running in time $\mathcal{O}(1.5^n)$ which finds the number of independent sets (or, equivalently, vertex covers) in a given n-vertex graph.

   You may need to prove that counting the number of independent sets in graphs of degree at most 2 is polynomial time solvable.

4. A *feedback vertex set* Z of graph G is a subset of vertices such that $G - Z$ is a forest.

   Show that if a graph on n vertices has minimum degree at least 3, then it contains a cycle of length at most $2\lceil \log n \rceil$. Use this to design a $(\log n)^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$-time algorithm for FEEDBACK VERTEX SET on undirected graphs. Is this an FPT algorithm for FEEDBACK VERTEX SET?

5. Let $\mathcal{F}$ be a set of graphs. We say that a graph G is $\mathcal{F}$-free if G does not contain any induced subgraph isomorphic to a graph in $\mathcal{F}$; in this context the elements of $\mathcal{F}$ are sometimes called forbidden induced subgraphs. For a fixed set $\mathcal{F}$, consider a problem where, given a graph G and an integer k, we ask to turn G into a $\mathcal{F}$-free graph by:

   (vertex deletion) deleting at most k vertices;

   (edge deletion) deleting at most k edges;

   (completion) adding at most k edges;

   (edition) performing at most k editions, where every edition is adding or deleting one edge.

   Considering $\mathcal{F}$ to be a fixed set means that $|\mathcal{F}| \in \mathcal{O}(1)$ and every graph in $\mathcal{F}$ has size $\mathcal{O}(1)$.

   Prove that, if $\mathcal{F}$ is finite, then there exists a $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$-time FPT algorithm for each of the four aforementioned problems. (Note that the constants hidden in the $\mathcal{O}()$-notation may depend on the set $\mathcal{F}$.)