Part A

Exercise 1 (Feedback Vertex Set in bipartite tournaments)
A *bipartite tournament* is an orientation of a complete bipartite graph. A *Feedback Vertex Set* of a directed graph G is a set of vertices S such that $G - S$ has no directed cycle.

1. Show that if a bipartite tournament has a directed cycle, then it has a directed cycle of length 4.

2. Detail an algorithm with running time $4^k n^{O(1)}$ that decides whether a bipartite tournament has a feedback vertex of size at most k. Justify the running time and the correctness of your algorithm.

Solution:
1. Let G be a bipartite tournament. Assume G has a directed cycle and let $C = (v_1, v_2, \ldots, v_k, v_1)$ be a smallest directed cycle G. Since G is bipartite, k is even. Assume for contradiction that $k \geq 6$. Since G is a bipartite tournament, there is an arc between v_1 and v_4. If $v_4 \to v_1$, then (v_1, v_2, v_3, v_4) is a directed cycle of length 4, contradicting the minimality of C. If $v_1 \to v_4$, then $(v_1, v_4, \ldots, v_k, v_1)$ is a directed cycle of length $k-2$, contradicting again the minimality of C.

2. The algorithm follows the same principle as the algorithm for vertex cover in slides 22.
We propose the following algorithm ALG for an instance (G, k).

1. If G has no directed cycle of length 4, return YES.

2. Otherwise, if $k = 0$, return NO.

3. Otherwise:

   - find a directed cycle (v_1, v_2, v_3, v_4) (this can be done in $O(n^4)$).
   - Compute, $ALG(G - v_i, k-1)$ for $i = 1, 2, 3, 4$ and return YES, if one of the $ALG(G - v_i, k-1)$ return YES.

   Each interior vertex of the search tree has degree 4, and the work done on each node of the search tree is $O(n^4)$, so the total running time is $4^k n^4$. The algorithm is correct because, by question 1 a set of vertices S of a complete bipartite tournament G is a FVS if and only if $G - S$ has no directed cycle of length 4.

Exercise 2 (Randomized FPT algorithm for Subgraph Isomorphism Problem)
In the Subgraph Isomorphism Problem we are given a graph G, a target graph H, where $|H| = k$, and we want to decide if G contains H as a (not necessarily induced) subgraph. We assume that H is connected, and that G has maximum degree d.

Design a randomised Monte-Carlo algorithm with one-sided error solving this problem in time $f(k, d) \cdot n^{O(1)}$ for some function f. Give an explicit formula for f. Justify the running time and the correctness of your algorithm.

Hint: Colour independently every edge of G in one of two colours, say red and blue, with probability $\frac{1}{2}$. Given a YES-instance G, say that such a colouring is *successful* if, naming $\hat{H}$ the copy of H in G, we get that every edge of $E(\hat{H})$ is coloured red, and every edge incident with a vertex of $\hat{H}$ but not in $E(\hat{H})$ is coloured blue.

Solution: The used technique is called "Random Separation", it is detailed in Section 5.3 of the book parametrized complexity.