Problem 1.

(a) **Base step.** Let \( n = 1 \). Since \( \sum_{i=1}^{1} f_i^2 = f_1^2 = 1 \) and \( f_1 f_2 = 1 \), then the assertion is true.

**Induction step.** Assume that \( \sum_{i=1}^{n} f_i^2 = f_n \cdot f_{n+1} \) for some \( n \geq 1 \). Then, using the induction hypothesis and the fact that \( f_{n+2} = f_{n+1} + f_n \), we obtain that

\[
\sum_{i=1}^{n+1} f_i^2 = \sum_{i=1}^{n} f_i^2 + f_{n+1}^2 = f_n \cdot f_{n+1} + f_{n+1}^2 = f_{n+1}(f_n + f_{n+1}) = f_{n+1} \cdot f_{n+2}
\]

(b) **Base step.** Let \( n = 1 \). Since \( f_2 f_0 - f_1^2 = 0 - 1 = -1 \) and \((-1)^1 = -1\), then the assertion is true.

**Induction step.** Assume that \( f_{n+1} \cdot f_{n-1} - f_n^2 = (-1)^n \) for some \( n \geq 1 \). Using the recursive definition \( f_{n+2} = f_{n+1} + f_n \), \( f_{n+1} = f_n + f_{n-1} \) and applying the induction hypothesis, we obtain

\[
f_{n+2}f_n - f_{n+1}^2 = (f_{n+1} + f_n)f_n - f_{n+1}(f_n + f_{n-1})
= f_{n+1}f_n + f_n^2 - f_{n+1}f_n - f_{n+1}f_{n-1}
= -(f_{n+1}f_n - f_n^2)
= (-1) \cdot (-1)^n
= (-1)^{n+1}.
\]

(c) **Base step.** Let \( n = 1 \). Observing that \[
\begin{bmatrix}
f_2 & f_1 \\
f_1 & f_0
\end{bmatrix}
= \begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix}
\]
and recalling the definition of \( A \), we easily check the equality.

**Induction step.** Assume that \( A^n = \begin{bmatrix}
f_{n+1} & f_n \\
f_n & f_{n-1}
\end{bmatrix} \). Then, using the recursive definition \( f_{n+2} = f_{n+1} + f_n \), \( f_{n+1} = f_n + f_{n-1} \) and applying the induction hypothesis, we obtain

\[
A^{n+1} = A^n A = \begin{bmatrix}
f_{n+1} & f_n \\
f_n & f_{n-1}
\end{bmatrix} \begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix}
= \begin{bmatrix}
f_{n+1} + f_n & f_{n+1} \\
f_n + f_{n-1} & f_n
\end{bmatrix}
= \begin{bmatrix}
f_{n+2} & f_{n+1} \\
f_{n+1} & f_n
\end{bmatrix}.
\]

Problem 2.

Let \( a_n \) be the answer to the problem and \( n = 3^k \). We have the following recursion : \( a_1 = 1 \), \( a_{3^k} = 3^k + a_{3^{k-1}} \), which follows immediately from the structure of the algorithm.

**Claim:** For all \( k \geq 0 \) we have that \( a_{3^k} = \frac{3^{2^k+1} - 1}{2} \).

**Proof by induction.**

**Base step:** \( a_{3^0} = 1 \). [And indeed, if \( n = 1 \), we print the sentence once.]
**Induction step:** Suppose that for \( k \geq 0 \) it is true that \( a_{3^k} = \frac{3^{k+1} - 1}{2} \). Then, using the recursion, we obtain \( a_{3^{k+1}} = 3^{k+1} + \frac{3^{k+1} - 1}{2} = \frac{3^{k+2} - 1}{2} \).

As a result, the phrase is printed \( \frac{3^{k+1} - 1}{2} = \frac{3n - 1}{2} \) times.

**Problem 3.** Let \( v \) be a vertex of the regular convex \( n \)-gon. The number of diagonals departing from \( v \) is \( n - 3 \), because there are \( n - 3 \) vertices which are not adjacent to \( v \). If we go over the \( n \) vertices and we sum the number of diagonals departing from each of them, we count every diagonal twice. Hence, the total number of diagonals is \( \frac{n(n - 3)}{2} \).

**Problem 4.**

(a) The first 3 letters are fixed, while the remaining 5 are free. Hence, this is equivalent to counting the strings of length 5 where each letter can take 26 different values. The total number of such strings is \( 26^5 \).

(b) We fix the first 2 and the last 2 letters, while the remaining 4 are free. Therefore, there are \( 26^4 \) possible strings.

(c) The number of strings which begin with \( ab \) is \( 26^6 \), the number of strings which end with \( yz \) is \( 26^6 \) and the number of strings which begin with \( ab \) and end with \( yz \) is \( 26^4 \). By inclusion-exclusion principle the number of strings which begin with \( ab \) or end with \( yz \) is \( 26^6 + 26^6 - 26^4 \).

(d) First of all, let us count the ways in which one can choose the positions of these four \( q \)'s. These are the combinations of 4 elements from a class of 12 without replacement, which means that there are \( \binom{12}{4} \) different ways of choosing the positions of the \( q \)'s. The remaining four positions are occupied by any of the 25 letters which is different from a \( q \). As a result, the total number of strings is \( \binom{12}{4} \cdot 25^4 \).

(e) **(Bonus.)** As concerns the first part of the question, counting the bit strings with exactly 4 1’s is equivalent to counting the ways in which one can choose the positions of these four 1’s. These are the combinations of 4 elements from a class of 12 without replacement. Therefore the solution is \( \binom{12}{4} \).

As concerns the second part of the question, let \( A \) be the set of bit strings of length 12 which have exactly four 1’s such that none of these 1’s are adjacent to each other. Let \( B \) be the set of bit strings of length 9 with exactly four 1’s. Consider a string \( a \in A \); it has four 1’s which are not adjacent to each other. Hence, each of the first three 1’s has to be followed by a 0. Consider the function \( f : A \to B \) which maps the string \( a \in A \) into the string \( b \in B \) such that \( b \) is obtained removing the three 0’s immediately after each of the first three 1’s of \( a \). It is easy to check that the function \( f \) is a bijection. Consequently, so \( |A| = |B| \). By reasons similar to those of point (d), \( |B| = \binom{9}{4} \) and, as a result, the solution is \( \binom{9}{4} \).