Question #1 [scores 2+3+4]. Given the symbols and their probabilities: \( p(a) = p(b) = 0.1, p(c) = 0.2, p(d) = p(e) = 0.11, p(f) = 0.38 \).

- Compute the Huffman code for this distribution.
- Compute the Canonical variant of the Huffman code (by sorting alphabetically the letters in every SYMB’s list).
- Decode the first 2 symbols of the coded sequence 11010...

Question #2 [scores 5]. Take your Matricola (of 6 digits), change every occurrence of 0 with 1 (if any), and then interpret each digit as an integer gap, and finally derive an increasing integer sequence by summing those gaps: namely, if the Matricola is 120304, then you transform it into 121314, and then you get the corresponding integer sequence as 1, 3 (=1+2), 4 (=1+2+1), 7 (=1+2+1+3), 8 (=1+2+1+3+1), 12 (=1+2+1+3+1+4).

- Compress the resulting increasing integer sequence with Elias-Fano.

Question #3 [scores 4+4]. Given the set of strings \( S = \{0000000, 0000010, 0001100, 0001110, 100, 1010\} \).

- Design a two-level storage scheme for \( S \) in which each disk page stores two strings which are Front-compressed, and the strings in internal memory are indexed via a Patricia Trie.
- Show how it is searched the string \( P = 000101 \)

Question #4 [scores 3+3+2]. Given the string \( T \) formed by your Matricola, and hence consisting of 6 digits:

- Show the suffix array of \( T \), in which every digit is interpreted as a symbol;
- Form the string \( P \) as given by the two middle digits of \( T \) (i.e. if the Matricola is 123456, then \( P = 34 \)). Then describe the algorithm that **efficiently counts** the occurrences of \( P \) in \( T \).
- Comment on the time complexity of the **counting** algorithm as a function of \( n \) (= \( T \)'s length), \( p \) (= \( P \)'s length) and the number \( occ \) of \( P \)'s occurrences.