CS 360 Online Quiz 2

CS 360 Online Quiz Honesty Statement
I am fully aware that once I access quiz problems I am allowed to work with my group partners only 90 minutes on them, that I may use the textbooks, lecture materials, and all other resources available via course website, but neither of the following is permitted: other books or materials, personal notes, web search tools, calculators, contacting other individuals outside my assigned group. By making a submission of my answers to the instructor I acknowledge that I followed the statement of online honesty.

You have 90 minutes for working out quiz problems, and still 15 more minutes for packaging your answers into pdf format and submitting them to the instructor via e-mail.

Solve all three problems. Each problem counts for 3 points. Extra credit problem also counts for three points.

1. Construct a non-ambiguous grammar generating the language consisting of all strings over the alphabet $\Sigma = \{0,1,2\}$, which contain no adjacent 1's. Provide a justification of correctness of your construction.
Hint: Adapt the grammar of figure 12 of weeks 1-2 file (the grammar on the left) to the current context. Introduce building blocks 0, 2, 10, 12 following the arrangements presented in figure 9 of weeks 1-2 file. Make sure to treat the ending symbol accordingly.

2. A Huffman tree constructed out of characters $a_1, a_2, a_3, \dots, a_n$, occurring with frequencies $f_1, f_2, f_3, \dots, f_n$, is given. Explain the terms: the maximal, the minimal, the average code length in terms of the features of the Huffman tree. Can it happen that the lengths are equal to 1, 2, 3,…,n-2, n-1, n-1? Can it happen that all code lengths are equal? Provide illustrating examples.
Hint: In the example of figure 2 of week 3 file the maximal code length in 4, the minimal code length is 2, the average code length is (2+2+2+3+4+4)/6. The depth of a node in a tree is its distance to the root.

3. Construct non-ambiguous grammars properly interpreting operator precedence, equivalent to $G: E \to E + E | E * E | (E) | id$, and satisfying the following properties

(i) both $+$ and $*$ are left associative,

(ii) both $+$ and $*$ are right associative,

(iii) operation $+$ is left associative and operation $*$ is right associative,

(iv) operation $+$ is right associative and operation $*$ is left associative.

Justify your constructions. In each case draw a parse tree of the string $1 * 2 * 3 + 4 * 5 * 6 + 7 * 8 * 9$. You are allowed to reuse portions of already drawn parse trees.

Hint: The grammar of figure 18 of weeks 1-2 file solves (i).

4. (Extra credit) Explain how one may construct an equivalent non-ambiguous grammar out of a given deterministic automaton. Provide illustrating examples.
Hint: Express the reading process of the automaton as a left-most derivation. Construct variables out of states.