Lab 03 - DFAs

Instructions:

- **A deterministic finite automaton (DFA) is a machine with no memory that recognizes languages.**
- **Your objective is to define a format DFA for each problem scenario. You must use the provided alphabet.**
- **A cumulative task will not receive credit if the required previous tasks are not completed.**
- **Your submissions must be submitted to the GitHub repository in the Lab03 directory.**
- **Cheating of any kind is prohibited and will not be tolerated.**
- **Violating or failing to follow any of the rules above will result in an automatic zero (0) for the lab.**

Grading

Task	Maximum Points	Points Earned
1	1.00	
2	1.30	
3	1.30	
4	1.40	
Total	5.00	

Note: solutions will be provided for tasks colored blue only.

Task 1

- A compiler recognizes an identifier token as a string containing one or more characters, each of which may be a lowercase letter (`L`), uppercase letter (`U`), underscore (`S`), or digit (`D`), that must begin with either a letter or an underscore. If the token starts with a digit or contains any invalid character (O) other than a letter, underscore, or digit, the compiler will report a lexical error.

  Create a text file named '`task01.txt`' that constructs a DFA that recognizes the language of strings over $\Sigma = \{\texttt{L}, \texttt{U}, \texttt{S}, \texttt{O}\}$ that are valid identifier tokens.

  Example:

  The strings `LLUD`, `DSU`, and `S` will be accepted, rejected, and accepted, respectively.

Task 2

- DNA (deoxyribonucleic acid) sequences are formed from four bases: Adenine (`A`), Guanine (`G`), Cytosine (`C`) and Thymine (`T`). In gene research, a motif is a short subsequence with biological significance.

  Create a text file named '`task02.txt`' that constructs a DFA that recognizes the language of DNA sequences over $\Sigma = \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$ that contain either the motif `G-A-T-C` or `C-T-A-G`.

  Example:

  The strings `AGACTAGC`, `AGTATAGT`, and `GATC` will be accepted, rejected, and accepted, respectively.

Task 3

- A binary number—a number composed only of 0s and 1s—is a multiple of 3 if the expression $2n + m$ is divisible by 3, where n and m are the number of 1s in the even and odd positions, respectively. Thus, the divisibility of a binary number by 3 depends on the positions of the 1s within the number.

  Create a text file named '`task03.txt`' that constructs a DFA that recognizes the language of binary numbers over $\Sigma = \{\texttt{0}, \texttt{1}\}$ that are multiples of 3.

  Example:

  The strings `10101`, `111`, and `11` will be accepted, rejected, and accepted, respectively.

Task 4

- A vending machine dispenses beverages priced \$1.00, \$1.50, \$1.75, or \$2.25. It accepts only quarters (`Q`) and dollars (`1`), and it requires exact change to dispense goods. If the customer selects an item (`S`) without having inserted the exact price or cancels (`C`), the machine returns the money and does not dispense (thus, restarting the process).

  Create a text file named '`task04.txt`' that constructs a DFA that recognizes the language of sequences of actions over $\Sigma = \{\texttt{Q}, \texttt{1}, \texttt{S}, \texttt{C}\}$ that will result in a beverage being dispensed.

  Example:

  The strings `1QQQS`, `1QS`, and `QC1S` will be accepted, rejected, and accepted, respectively.

Extra Credit

- Create a text file named '`extra.txt`' that constructs a DFA that recognizes the union of the languages

$$L_1 = \{w : \texttt{011} \notin w\}$$
$$L_2 = \{w : w \text{ is one or more 0s between one or more 1s}\}$$

  (0.5 points)