**Compiler project**

**Phase 1 lexical analyzer**

**Team members:**

**Ahmed Talaat Noser (6)**

**Salma Ragab Gad (32)**

**Problem statement:**

This phase of the assignment aims to practice techniques for building automatic lexical

analyzer generator tools.

**Data structures used:**

* Unordered maps to store the transition between states with integer key represents the first state and unordered map value represents the input and next state.
* Vectors of “RegexExp” to store expressions, definitions, keywords and punctuations where RegexExp is a struct type defines a string for the type or the name of expression and another one for the expression itself.
* Disjoint sets used in the minimization of the DFA.

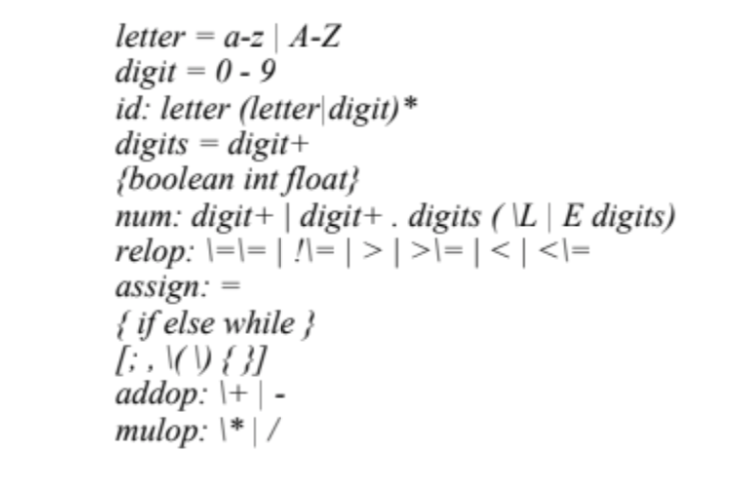
**Algorithms and techniques used:**

* **Thompson’s construction:** to construct the NFA of a regular expression based on basis expressions and basis operators.
* **Subset construction:** to construct the DFA from the combined NFA starting with adding an unmarked DFA state that represents the E-closure of the start state of the combined NFA then, continue to add unmarked DFA states that represents the E-closure of the reachable NFA states from each unmarked DFA states then marking them until no unmarked DFA states are left.
* **Minimize DFA algorithm using Union Find:** using partitioning method, we start by a partition that contains two groups (final, non-Final states) then, trying to create new partition by further partitioning each group of the old partition if we find non-distinguishable in it until the number of groups in the new partition equals the number of groups in the old partition.
* **Pattern matching using maximal munch and priority:** we use the DFA transition table to match a character by character of the input code until we stuck (reach the empty state) and then, pick the last match which will be the longest matched prefix and return the token type from the associated accepting state.

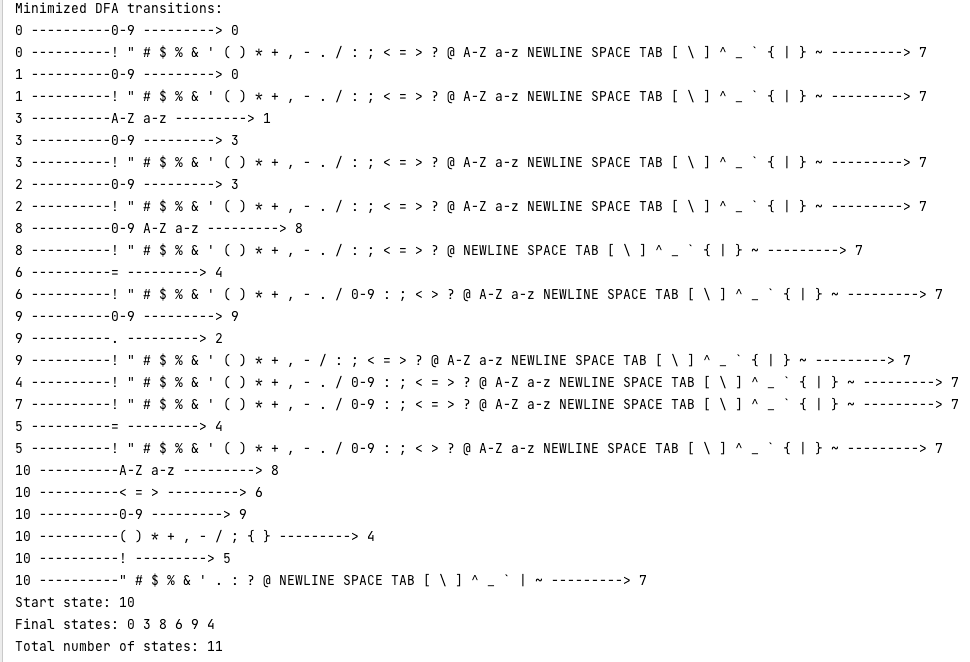
**Assumptions:**

* The character (~) is reserved for concatenation (i.e. must be escaped in rules file).
* Epsilon is represented by 0 (null character).
* The allowed input character set is the whole ASCII set.

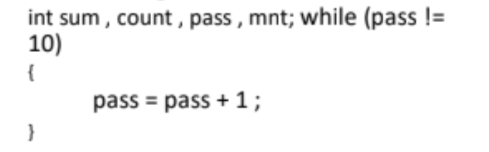
**Input file example:**

****

**Transition table of minimized DFA:**

****

**Test program:**

****

**Output file for the test program:**

