COMP 370, Spring 2018
Program #1, DFA Simulation
Date Assigned: Tuesday, February 13
Date Due: Saturday, Feburary, 24 , 3AM
Possible Points: 20

Write a program that simulates the computation of a DFA on a series of input strings, and reports the results of those computations. Your program should behave exactly as described below.

The program should read input from a file that is specified as the command line argument to the program, and write output to standard output. Input consists of a description of the DFA to simulate, followed by a series of strings for which the DFA's computation should be simulated.

The format of the DFA description is as follows:

1. An integer that is the number of states in the DFA. This appears by itself on the first line of input. (In the remainder of the description, each state must be referred to by an integer in the range $[1, 2, ..., n]$, where n is the number of states.)

2. The alphabet of the DFA. This appears by itself on the second line of input. Every character in the line (not including the terminating newline character) is a symbol of the alphabet.

3. The transition function of the DFA. There will be one line of input per entry in the transition function table, starting with the third line of input. The format of an entry is

   ```
   qa 'c' qb
   ```

   This entry indicates that if the DFA is in state qa and the next symbol scanned is a c, then the DFA transitions to qb.

   qa and qb must be valid states; that is, qa $\in \{1, 2, \ldots, n\}$ and qb $\in \{1, 2, \ldots, n\}$. c must be in the alphabet, and the single quotes must be present.

   The entries of the transition function can appear in any order.

4. An integer that is the start state of the DFA. This appears by itself on the first line after the transition function lines.

5. The set of accept states of the DFA. These appear together on the line following the line containing the start state.

Multiple entries on a line are separated by 1 or more whitespace characters. The first entry on a line is preceded by 0 or more whitespace characters, and the last entry on a line is followed by 0 or more whitespace characters (not counting the newline character).

Following the DFA specification are the string inputs to be simulated on the DFA, one string per line. All characters on each line (except the terminating newline character) are part of the input string. The input strings should start on the line following the last line of the DFA specification. They end with the last line of the file.

Your program can assume that the input file will be correctly formatted, with no inconsistent data. (For example, if there are only 6 states in the DFA, there will be no transition function entries that specify a state of greater than 6 or less than 1.)

Your simulation program should output one line per input string, and the line should contain "Accept" if the DFA accepts the input string, and "Reject" if the DFA rejects the input string. (The quotes are not part of what you output.) No other characters should appear on a line of output, and there should be no other lines of output.

You can program in Java, C, C++, or Python. Follow the programming style given in the document on blackboard (under the "Programming Style Guidelines" link).

You should work with a programming partner, using the pair programming software development technique.

Your program should be contained in one source file called `pa1` (with the appropriate extension for the language you are using). Do not submit more than one file, or an entire folder.

Test inputs and correct outputs are in a compressed testcases folder on the assignment webpage. Your program should work correctly on all of these tests.

Here is one of them (dfa1.txt):

```
6
01
1 '0' 4
1 '1' 2
2 '0' 4
2 '1' 3
3 '0' 3
3 '1' 3
4 '0' 4
4 '1' 5
5 '0' 4
5 '1' 6
6 '0' 4
6 '1' 6
1
3 6
```

```
010101010
111011110
01110011
11111
01010000
```

Correct output for the sample input (correct1.txt) is

```
Reject
Accept
Accept
Accept
Reject
```

To submit your program log into the class website (blackboard), go to Programming Assignments, click on PA1, click on Browse My Computer under Attach File and upload your pa1 file. Once your file has been successfully uploaded, click on Submit. Your program is now submitted.