Homework Assignment #2
Due: September 22, 2016 at 4:00 p.m.

Submission Instructions: Each question below requires you to design a finite automaton. You must type a description of your automaton (using the file format described below). Use separate files for each automaton and label the files `q1.txt` and `q2.txt`. See the course web page for instructions on using the submit command.

If you are working with a partner, only one of you should submit your solutions. Submit a separate file called declaration.txt that gives the name and student number of your partner. ("My partner is ...")

Note: there is a Java programme called `DFA.java` posted on the course web page that allows you to test your solutions before you submit them. Read the comments in `DFA.java` to see how the programme works.

1. Consider the input alphabet $\Sigma = \{0, 1\}$. Design a deterministic finite automaton that accepts a string if and only if its last character is different from its third-last character.

For example, the string `111010` should be rejected because the third-last character and the last character are both `0`. The string `110` should be accepted because the third-last character is `1` and the last character is `0`. The string `00` should be rejected because it does not have a third-last character. (All strings of length less than three should be rejected.)

In the comments after the description of your automaton, explain in English what each state of your automaton represents (that is, describe which strings take your automaton to that state).

2. The Leutonian alphabet consists of 4 letters: $\Sigma = \{$`a, b, m, o`$\}$. The characters `a` and `o` are vowels, while `b` and `m` are consonants. There are two spelling rules that all Leutonian words must follow:

1. vowels and consonants must alternate (i.e., there can never be two consonants or two vowels side by side), and

2. an `o` can never appear later in the word than an `a`.

E.g., `mama` and `obama` satisfy Leutonian spelling rules, but `mamma` and `babomo` do not.

Design a deterministic finite automaton that accepts a string if and only if it follows the two rules of Leutonian spelling.

In the comments after the description of your automaton, explain in English what each state of your automaton represents.

York University Finite Automaton File Format (YUFAFF)

The YUFAFF is a way to describe a finite automaton in a text file. It can be used for deterministic or non-deterministic automata.

Suppose you wish to describe an automaton $(Q, \Sigma, \delta, q_0, F)$. We shall assume the state names are $q_0, q_1, \ldots, q_{n-1}$, where q_0 is the starting state. (If this is not the case, just rename the states.)

The first line of the file contains four positive integers n, m, k and t separated by single spaces, where $n = |Q|, m = |\Sigma|, k = |F|$ and t is the number of lines in the file that will be used to describe transitions of the automaton.

The second line of the file contains m strings $\sigma_0, \sigma_1, \ldots, \sigma_{m-1}$ separated by single spaces. These strings are the names of the characters of the input alphabet. These strings should be distinct and should not contain any spaces. Also, the string `empty` is not allowed as a character name. (Ordinarily, you will use single-character strings to name the characters, but we allow strings more generally in case you want to talk about an automaton that uses an alphabet of non-ASCII input characters. For example, if your input alphabet was $\Sigma = \{\alpha, \beta, \gamma\}$, you might use the names `alpha beta gamma` for the characters.)

The third line contains k natural numbers separated by single spaces. These numbers are the indices of the accepting states. For example, if the accepting states are q_0, q_6 and q_9 then the third line of the file should contain `0 6 9`.

The following t lines each contain the description of one transition defined by δ.

For *deterministic* finite automata, each of these t lines contains a natural number i, a character name x and another natural number j, separated by single spaces. There should be one line containing i x j if and only if $\delta(q_i, x) = q_j$. The values should satisfy $0 \leq i < n$, $0 \leq j < n$, $x \in \{\sigma_0, \ldots, \sigma_{m-1}\}$. (For deterministic finite automata, we should have $t = mn$.)

For *non-deterministic* finite automata, each of the t lines contains a natural number i, a string x and another natural number j, separated by single spaces. The string x should either be a character name in $\{\sigma_0, \ldots, \sigma_{m-1}\}$ or should be the string `empty` to indicate a ε-transition. There should be one line containing i x j if and only if $q_j \in \delta(q_i, x)$.

Anything after these t lines will be ignored, so you can use them to write comments or explanations.

Example: The deterministic automaton shown in Figure 1.14 of the textbook could be described as follows in YUFAFF:

```
3 4 1 12
<RESET> 0 1 2
0
0 <RESET> 0
0 0 0
0 1 1
0 2 2
1 <RESET> 0
1 0 1
1 1 2
1 2 0
2 <RESET> 0
2 0 2
2 1 0
2 2 1
Comment: This file describes the automaton in Figure 1.14 of the textbook
```