

Assignment 3 is due at **5 p.m. Tuesday, February 20**.

## Instructions

Write programs as described below. **E-mail** your work to `ffggc@uaf.edu` with subject “AA3”. Source (and, if required, data) files should be **attached** to your e-mail message.

*Note: I may not read your homework e-mail immediately. If you wish to discuss the assignment (or anything else) with me, send me a separate message with a different subject line.*

## Exercises

### Exercise 1

Write a program that simulates a DFA that recognizes Roman numerals.

A *Roman numeral* is a sequence of upper-case letters, possibly including M, D, C, L, X, V, and I, representing an integer from 1 to 3999, according to the usual rules.

Your code should conform to the following requirements.

- The language you use can be C, C++, or any other language approved **in advance** by your instructor.
- The input should be taken from a file whose name is entered by the user after the program is run (no command-line parameters). You may assume that the input contains only upper-case letters and newlines.
- The DFA should be run once on each line of the input file. For each line, print “YES” if the DFA accepts the line, or “NO” otherwise.
- The code must actually simulate a DFA. That is: You must keep track of the current state. You must use *only* the current state and one input symbol to determine the next state. You must use *only* the state at the end of the line to determine what to print.
- For the input alphabet of your DFA, you may use either the set of all upper-case letters or the set of all upper-case letters and an end-of-line marker, whichever you find more convenient.
- Your code must be neat and readable, and it should be clear to the reader that you are simulating a DFA, how the state is stored, how transitions are done, and how acceptance is tested.

For example, suppose your program is given the following input:

MCMLXXXIV  
MIX  
HOWDY  
III  
IIII  
IV

Then the output should be:

YES  
YES  
NO  
YES  
NO  
YES

## Exercise 2

Like exercise 1, but the DFA should recognize “reasonable words”. A *reasonable word* is a string of upper-case letters in which

- no two consecutive letters are both vowels,
- no three consecutive letters are all consonants,
- the first two letters are not both consonants,
- the last two letters are not both consonants, and
- there is at least one vowel.

The vowels are A, E, I, O, and U. The other 21 letters are consonants.

For example, suppose your program is given the following input:

AUTOMATA  
AND  
FORMAL  
LANGUAGES  
IS  
A  
NIFTY  
CLASS

Then the output should be:

NO  
NO  
YES  
NO  
YES  
YES  
NO  
NO

### **Extra Credit Option**

For extra credit, instead of writing two programs, write just one program that reads a description of a DFA from a data file, and then executes the DFA on some input file.

If you do this option, send me one program, two data files, and (brief!) instructions on how to run the program.