YOUR NAME: REGISTRATION #

(J) Lexicondensed (1/4) [15 points]

Compiling a lexicon (a catalog of words) can be time-consuming and difficult because each individual word has so many potential forms. Suppose that you are dealing with the following words:

> view, viewed, viewing, views, review, reviewed, reviewing, reviews, watch, watched, watches, watching, rewatch, rewatches, rewatching, rewatched, wave, waved, waves, waving, rewave, rewaves, rewaved, and rewaving.

Writing all of these forms is tedious; even though you generate a list, you will probably feel listless. Therefore, instead of using this brute force method, you can condense the list with the format shown below:

```
VERBPREFIX      VERBSTEM        VERBSUFFIX
re              watch           ed
Ø               view            s
                wave            ing
                                Ø
```

This setup generates a list of all words that consist of one component of `VERBPREFIX` followed by one component of `VERBSTEM` followed by one component of `VERBSUFFIX` (the Ø stands for an empty spot, so a word could have no letters in the `VERBPREFIX` or `VERBSUFFIX` slot). The list generated is identical to the brute force list but is much less tedious to create.

There is one major problem, however. The way that this format strings together word components (called morphemes) does not account for spelling changes that may occur along the way. For example, many legitimate words are generated, such as watch, review, and rewaves, but some misspelled words also result, such as watchs and waveing. In order to fix this, you also need to write a set of spelling change rules to describe these changes. The applicable rules in this case are:

```
ch -> che || * s
e -> Ø || * [ed | ing]
```

These rules mean "ch turns into che if ch is followed by s" and "e turns into nothing if e is followed by ed or ing."

There are many different ways that this type of rule can be written. Here are a few more examples of such rules and their meanings:

`u -> w || * Vowel`	(u turns into w if u is followed by a vowel)
`np -> mp`	(np always turns into mp)
`t -> c || Consonant * kf`	(t turns into c if it is between a consonant and kf)
`[l | f | r] -> z || w * [c | p]`	(each letter l, f, or r will turn into z if it falls between w and either c or p)

YOUR NAME: REGISTRATION #

(J) Lexicondensed (2/4)

J1. Consider the following lexicon and set of rules. (Note that the rules apply in the order given).

```
PARTONE          PARTTWO          Spelling Change Rules:
cdn              rgt              vsk -> ko
cav              sks              nbj -> jirj
_ _ _            _ _ _            nsk -> jeej
                                  gt -> e || avr *
                                  j -> res || avb *
                                  j -> tu || b *
                                  gt -> ar
                                  vb -> yp
                                  cdj -> b
                                  c -> cal || q * y
                                  js -> ch
                                  os -> o || ak *
                                  ak -> jinkcj || c *
                                  cj -> g
                                  dnr -> ed
                                  s -> ry || o *
                                  q -> hi || * ck
                                  q -> eu || * ca
                                  ay -> y || * p
                                  qc -> po
                                  c -> m || * av
                                  vr -> pl
```

A. Write the four words generated by the above lexicon and set of rules.

B. If you add two more three-letter entries to the lexicon (one entry in `PARTONE` and one entry in `PARTTWO`), the system will generate an additional five words that go together with the four words from Task 1. What are the new entries for `PARTONE` and `PARTTWO`?

What are the five newly generated words? *(Hint: Every rule is used at least once.)*

YOUR NAME: REGISTRATION #

(J) Lexicondensed (3/4)

J2. The following lexicon and incomplete set of spelling change rules was designed to output a list of adjectival forms of country names, as shown in the table on the next page. It works almost exactly as intended: the output of the setup is identical to the "Desired Adjective" column (on the next page) *except* that it produces the wildly incorrect word "ottruese" in place of "australian."

TASK: Fill in the blanks in the Spelling Change Rules (just write your answers directly in the blanks in the box below). Each blank stands for a single letter. Remember that these rules will produce "ottruese" instead of "australian" and that the rules apply in the order given.

```
COUNTRY        ENDING    Spelling Change Rules
andorra        ian
australia                _ _ -> _ _ _
bhutan
bolivia                  ian -> Ø || t *
cambodia
chad                     _ _ -> Ø
chile
china                    _ _ -> Ø|| [ _ i | e _ i ] *
congo
cuba                     _ _ _ -> _ _ _ _
cyprus
england                  i -> Ø || [ _ | _ | _ | a _ | _ ] *
fiji
guyana                   _ _ -> Ø || c *
indonesia
israel                   _ _ _ -> _ _ _ _ || _ *
japan
kenya                    _ _ _ i a n -> _ _ _
mexico
morocco                  _ _ -> _ _ _ || [ n | m ] *
nauru
netherlands              _ _ _ _ _ _ _ _ _ _ _ _ _ _ -> _ _ _ _ _
poland
portugal
rwanda
singapore
sudan
togo
uganda
vietnam
yemen
```

YOUR NAME: REGISTRATION #

(J) Lexicondensed (4/4)

Country	Desired Adjective
andorra	andorran
australia	australian
bhutan	bhutanese
bolivia	bolivian
cambodia	cambodian
chad	chadian
chile	chilean
china	chinese
congo	congolese
cuba	cuban
cyprus	cypriot
england	english
fiji	fijian
guyana	guyanese
indonesia	indonesian
israel	israeli

Country	Desired Adjective
japan	japanese
kenya	kenyan
mexico	mexican
morocco	moroccan
nauru	nauruan
netherlands	dutch
poland	polish
portugal	portuguese
rwanda	rwandan
singapore	singaporean
sudan	sudanese
togo	togolese
uganda	ugandan
vietnam	vietnamese
yemen	yemeni