PROLOG LAB, CS440, 25 points. DUE: September 30.

ASSISTANTS such as Alexa and Siri use a combination of machine learning as well as parsing.
Here we explore what can be done with simple template matching.

“mymatch" determines if two lists are the same. mymatch allows two types of variables: keyword and mwords.

> Keyword must match exactly one word in a list.
> Mwords can match multiple words in a list. It can also match an empty list.

NOTE: The “variables” keyword and mwords in this lab are actually not Prolog variables, but rather are atomic words that your program should flag as special “wildcards” that match other words.

The first list is just a list of words.
The “wildcards” appear in the second list.
The binding of these wildcards is returned as sublists of the parameter "Binding."

FORMAT: ? mymatch(WordList, VariableList, Binding).

I will break up the examples, from simpler to more complex.

ONE) In its simplest form, mymatch will just determine if two lists are the same.

```
? mymatch( [these, sentences, are, the, same], [these, sentences, are, the, same], Binding).

Yes.
Binding = [[]].
```

TWO) Next, mymatch should match sentences using one “keyword” wildcard.

```
? mymatch( [siri, play, jazz], [siri, play, keyword], Binding).

Yes.
Binding = [[keyword, jazz]].                  /* NOTE:  the word “keyword” must be included with its “binding” */
```

THREE) Next, mymatch should match sentences using one “mwords” wildcard.

```
?mymatch([siri, play, kind, of, blue, please], [siri, play, mwords, please], Binding).

Yes.
Binding = [[mwords, kind, of, blue]].     /* NOTE:  the word “mwords” must be included with its “binding” */

? mymatch([the, first, or, second, example], [the, keyword, example], Binding).

No.

? mymatch([the, first, or, second, example], [the, mwords, example], Binding).

Yes.
Binding = [[mwords, first, or, second]].
```

FOUR) In its most general form, mymatch should match sentences using any combination that uses “keyword” and “mwords” wildcards multiple times.

```
? mymatch([one, two, three, four, five, six, seven, eight],
          [one, mwords,     four, mwords, seven, keyword], Binding).

Yes.
Binding = [[mwords, two, three], [mwords, five, six], [keyword, eight]].

? mymatch([siri, first, play, something, slow, and, then, play, something, fast]
          [siri, first, play,  mwords,            and,  then, play, mwords, keyword], Binding).

Yes.
Binding = [[mwords, something, slow], [mwords, something], [keyword, fast]].


? mymatch([siri, first, play, something, slow, and, then, play, jazz, please],
          [siri, first, play,  mwords, and, then, play, mwords, keyword, please], Binding).

Yes.
Binding = [[mwords, something, slow], [mwords, []], [keyword, jazz]].


? mymatch([one, two, zee,    for, fat, six,      set,    ate,  nil,    ten]
          [one, mwords,      for, mwords,  set, keyword, nil, keyword],  Binding).

Yes.
Binding = [[mwords, two, zee], [mwords, fat, six], [keyword, ate], [keyword, ten]].


? mymatch([one, two, zee,  for, fat, six,    set,    ate,       nil,    ten,    last],
          [one, mwords,  for, mwords, set, keyword,  nil, keyword], Binding).

No.


? mymatch([siri, add, coffee, to, the, grocery, list, and, play, a café, song]
          [siri, add, mwords, grocery, list, mwords], Binding).

Yes.
Binding = [[mwords, coffee, to, the], [mwords, and, play, a, cafe, song]].


? mymatch([siri, add, coffee, to, the, grocery, list, and, play, a, café, song]
          [siri,  mwords, and, play, mwords], Binding).

Yes.
Binding = [[mwords , add, coffee, to, the, grocery, list], [mwords, a, cafe, song]].
```