Problem 1. (*Spell Checker*) Write a program `spell_checker.py` that accepts words from standard input; looks up each word in the file `data/misspellings.txt` that maps misspelled words to their correct spellings; and if it exists (ie, is misspelled), writes the word to standard output along with the correct spelling.

>_ ~/workspace/exercise6

```
$ python3 spell_checker.py
Try nto to become a man of sucess but rather try to become a man of value. ~ Albert Einstein
<enter>
nto -> not
sucess -> success
<ctrl-d>
```

spell_checker.py

```
from instream import InStream
from symboltable import SymbolTable
import stdio


# Entry point.
def main():
    # Set inStream to an input stream built from the file 'data/misspellings.txt'.
    ...

    # Set lines to the list of lines read from inStream.
    ...

    # Set misspellings to a new symbol table object.
    ...

    for ... in ...:
        # For each line (of the form 'misspelling correction') in lines...

        # Set tokens to the list obtained by splitting line using the split() method from str.
        ...

        # Insert the pair tokens[0]/tokens[1] into misspellings.
        ...

    while ...:
        # As long as standard input is not empty...

        # Set word to a string read from standard input.
        ...

        # If word exists in misspellings, then it is misspelled. So write the word and the
        # correction to standard output, separated by the string '->'.
        ...


if __name__ == '__main__':
    main()
```

Problem 2. (*Word Occurrences*) Write a program `word_occurrences.py` that accepts *filename* (str) as command-line argument and words from standard input; and writes to standard output the word along with the indices (ie, locations) where it appears in the file whose name is *filename* — writes "Word not found" if the word does not appear in the file.

>_ ~/workspace/exercise6

```
$ python3 word_occurrences.py data/Beatles.txt
dead
<enter>
dead -> [3297, 4118, 4145, 4197]
parrot
<enter>
Word not found
<ctrl-d>
```

word_occurrences.py

```
from instream import InStream
from symboltable import SymbolTable
import stdio
import sys
```

```
# Entry point.
def main():
    # Accept filename (str) as command-line argument.
    ...

    # Set inStream to an input stream built from filename.
    ...

    # Set words to the list of strings read from inStream.
    ...

    # Set occurrences to a new symbol table object.
    ...

    for i, word in enumerate(...):
        # For each word (having index i) in words...

        # If word does not exist in occurrences, insert it with an empty list as the value.
        ...

        # Append i to the list corresponding to word in occurrences.
        ...

    while ...:
        # As long as standard input is not empty...

        # Set word to a string read from standard input.
        ...

        # If word exists in occurrences, write the word and the corresponding list to standard
        # output, separated by the string '->'. Otherwise, write the message 'Word not found'.
        if ...:
            ...
        else:
            ...


if __name__ == '__main__':
    main()
```

Files to Submit

1. `spell_checker.py`
2. `word_occurrences.py`

Before you submit your files, make sure:

- You do not use concepts from sections beyond "Stacks, Queues, and Symbol Tables".
- Your code is adequately commented, follows good programming principles, and meets any specific requirements such as corner cases and running times.