Problem 1. (*Array-based Symbol Table*) Implement a data type called `ArrayST` that uses an unordered array as the underlying data structure to implement the basic symbol table API.

```
>_ ~/workspace/exercise5
$ java ArrayST < data/tinyST.txt
S 0
E 12
A 8
R 3
C 4
H 5
X 7
M 9
P 10
L 11
```

Problem 2. (*Spell Checker*) Implement a program called `Spell` that accepts *filename* (String) as command-line argument, which is the name of a file containing common misspellings (a line-oriented file with each comma-separated line containing a misspelled word and the correct spelling); reads text from standard input; and writes to standard output the misspelled words in the text, the line numbers where they occurred, and their corrections.

```
>_ ~/workspace/exercise5
$ java Spell data/misspellings.txt < data/war_and_peace.txt
wont:5370 -> won't
unconciousness:16122 -> unconsciousness
accidently:18948 -> accidentally
leaded:21907 -> led
wont:22062 -> won't
aquaintance:30601 -> acquaintance
wont:39087 -> won't
wont:50591 -> won't
planed:53591 -> planned
wont:53960 -> won't
Ukranian:58064 -> Ukrainian
wont:59650 -> won't
conciousness:59835 -> consciousness
occuring:59928 -> occurring
```

Files to Submit

1. `ArrayST.java`
2. `Spell.java`

Before you submit your files, make sure:

- You do not use concepts outside of what has been taught in class.
- Your code is adequately commented, follows good programming principles, and meets any specific requirements such as corner cases and running times.