Assignment 3

Out: August 8, 1999
Due: August 22, 1999
Extension: August 26, 1999

1 IntQueue

Modify the `peek` and `unqueue` methods in your `IntQueue` class so that they throw exceptions when the queue is empty. (And fix whatever bugs you have.) Specifically, they now look like:

- `public int peek() throws IQEmptyException`
- `public int unqueue() throws IQEmptyException`

`IQEmptyException` is defined as follows (or you can download it from the course webpage).

```
public class IQEmptyException extends Exception
{
  public IQEmptyException() {
    super();
  }
  public IQEmptyException(String m) {
    super(m);
  }
}
```

In addition, put `IntQueue` and `IQEmptyException` under package `intqueue`. Write a demo program under the anonymous package.

2 Word Search

Write a program to search words in the Scrabble way. Specifically, the program accepts a filename *dict* and a string s of letters as command line arguments. Here *dict* is the name of a dictionary file, sorted alphabetically; and s represents the letter tiles you can use. The program then reads *dict* and look for words that use only the letter tiles found in s, i.e., if s contains only n copies of a letter, then the word must contain at most n copies of it. The search is case-sensitive, so that by using lower case letters in s we can avoid proper nouns. The program outputs the found words. A typical *dict* would be `/usr/dict/words`.

A typical run of the program would be

```
java Search /usr/dict/words queuen
```

and the corresponding output would be

```
en
enqueue
nu
queen
queue
```

Note that "nun" is not output because the available tiles have only one "n".

You are encouraged to discuss with each other on the search algorithm.

Do not use `SimpleInput`. Handle exceptions yourself.