UNDER CONSTRUCTION

This assignment is optional. Its sole purpose is to ensure that you can write simple Java programs, use `algs4.jar`, and submit them to the Coursera autograder.

0. **Install our Java programming environment (optional).** Install our novice-friendly Java programming environment on your computer by following these step-by-step instructions for Mac OS X , Windows , or Linux . On each assignment, use the Project from the menu at top.

   As part of these instructions, you will write, compile, and execute the program HelloWorld.java .

```
~/Desktop/hello> javac HelloWorld.java

~/Desktop/hello> java HelloWorld
Hello, World
```

1. **Command-line arguments.** Write a program `HelloGoodbye.java` that takes two names as command-line arguments and prints hello and goodbye messages as shown below (with the names for the hello message in the same order as the command-line arguments and with the names for the goodbye message in reverse order).

```
~/Desktop/hello> javac HelloGoodbye.java

~/Desktop/hello> java HelloGoodbye Kevin Bob
Hello Kevin and Bob.
Goodbye Bob and Kevin.

~/Desktop/hello> java HelloGoodbye Alejandra Bahati
Hello Alejandra and Bahati.
Goodbye Bahati and Alejandra.
```

2. **Using algs4.jar.** *Under construction.* Write a program `RandomWord.java` that reads a sequence of words from *standard input* and prints one of those words uniformly at random. Do *not* store the words in an array or list. Instead, use *Knuth's method*: when reading the ith word, select it with probability $1 / i$ to be the champion, replacing the previous champion. After reading all of the words, print the surviving champion.

```
~/Desktop/hello> javac-algs4 RandomWord.java

~/Desktop/hello> java-algs4 RandomWord
heads tails
tails

~/Desktop/hello> java-algs4 RandomWord
heads tails
heads

~/Desktop/hello> more animals8.txt
ant bear cat dog
emu fox goat horse

~/Desktop/hello> java-algs4 RandomWord < animals8.txt
emu

~/Desktop/hello> java-algs4 RandomWord < animals8.txt
bear
```

Use the following library functions from `algs4.jar` :

- `StdIn.readString()` : reads and returns the next string from standard input.
- `StdIn.isEmpty()` : returns `true` if there are no more strings available on standard input, and `false` otherwise.
- `StdOut.println()` : prints a string and terminating newline to standard output. It's also fine to use `System.out.println()` instead.
- `StdRandom.bernoulli(p)` : returns `true` with probability p and `false` with probability $1 - p$.

In order to access these library functions, you must do the following two things:

- Add `algs4.jar` to the *Java classpath*. This typically requires a different mechanism from the command line and the IDE.
  - If you used our autoinstaller, the *Bash* commands `javac-algs4` and `java-algs4` add `algs4.jar` to the Java classpath.
  - If you use *IntelliJ*, the supplied *IntelliJ* project folder includes `algs4.jar` and adds it to the Java classpath.
  - If you prefer to use some other shell (such as *Powershell* or *zsh*) or IDE (such as *Eclipse* or *Netbeans*), that's fine—just be sure that you can configure it accordingly.
- Add an `import` statement like the following at the top of your program:

```
import edu.princeton.cs.algs4.StdIn;
import edu.princeton.cs.algs4.StdOut;
import edu.princeton.cs.algs4.StdRandom;
```

If you use *IntelliJ* and the provided project folder, *IntelliJ* will automatically add and remove `import` statements as needed, so you won't need to type them.

Web submission. Submit a ZIP file containing only `HelloWorld.java`, `HelloGoodbye.java`, and `RandomWord.java`. Your submission may not call library functions except those in `java.lang` and the ones in `algs4.jar` enumerated

above.