

CMPS 12A
Introduction to Programming
Spring 2007

Programming Assignment 3
Due Wednesday May 23, 10:00 pm

This project will be based on your solution to lab4, so it would be a good idea to review the specifications for that assignment and fix any remaining bugs (see lab5 in this regard.) Recall your lab4 program read a single integer from standard input and tested it for primality. Your program in this assignment will be called `Primes.java`, and will read any number of positive integers from a file, do a primality test on each, then print a result for each integer to standard output. If the number is composite, your program will print its smallest prime divisor. Before you begin work on this assignment, you should study the examples `Echo.java`, `FileIO.java`, `TryCatch.java`, `ReadIntegers.java`, and `Cat.java`, recently added to the webpage, to learn about command line arguments, reading and processing input from a file, and catching exceptions. Your program will be given the name of its input file via a command line argument, as follows:

```
% java Primes input_file
```

It will read only positive integers from this input file, skipping over any zeros, negative integers, or non-integer strings. Program output will be printed to standard out, formatted as in the following example.

Input file `numbers1.dat` (see webpage examples for this and other files):

```
-2 -1 0 1 2 3 4 5 6 7 8 -9 10 11 12 13 14 15 16 17 xxx 18 19 20
```

Output:

```
% java Primes numbers1.dat
1 is neither prime nor composite
2 is prime
3 is prime
4 is divisible by 2
5 is prime
6 is divisible by 2
7 is prime
10 is divisible by 2
11 is prime
12 is divisible by 2
13 is prime
14 is divisible by 2
15 is divisible by 3
16 is divisible by 2
17 is prime
18 is divisible by 2
19 is prime
20 is divisible by 2
```

Observe that for purposes of this assignment, 1 is categorized as neither prime nor composite. If too many or too few command line arguments are given, your program will respond with a usage message, then quit.

```
% java Primes aaa bbb
Usage: java Primes input_file
```

If the file specified on the command line does not exist, or is not accessible for some reason, your program will print the system message associated with `FileNotFoundException`, print the same usage message as before, then quit.

```
% java Primes xxx
xxx (The system cannot find the file specified)
Usage: java Primes input_file
```

Your program will define two static functions called `testPrime()` and `Usage()` with the following signatures.

```
static int testPrime(int n)
static void Usage()
```

Method `testPrime()` will return 1 if its argument n is prime, it will return the least prime divisor of n if n is composite, and it will return 0 if $n \leq 1$, indicating that n is neither prime nor composite. The operation of this function should be based on your main loop in lab assignment 3. Just as in that assignment, your program should only do trial divisions of n by numbers d in the range $2 \leq d \leq \sqrt{n}$. Note that if n is composite, the first successful trial division in this process will be by the least prime divisor of n , which should then be returned by the function. Method `testPrime()` will be called from `main()`, and its return value will be used to determine what to print to the screen. Method `Usage()` will print the above usage statements to standard error, then quit. Follow the example `ReadIntegers.java` on the webpage.

What to turn in

Compile and test your program `Primes.java` on the input files `numbers1.dat`, `numbers2.dat`, and `numbers3.dat` found on the webpage. Submit your source file to the assignment name `pa3`, then verify the submission in the usual ways.