

<http://www.cs.arizona.edu/classes/cs245/spring08/>

Homework #8

(44 + 1 + 5 points)

Due Date: April 11th, 2008, at the beginning of class

Name: _____ Section Leader: _____

Directions: Write complete answers to each of the following questions. All are from the ends of the indicated sections in our text; for these, provide complete answers in accordance with the directions given. Show your work, when appropriate, for possible partial credit. This is not a group project; do your own work. The section leaders will go over the homework problems with you during the section meeting following the due date. If you need help, *remember that the section leaders and I have office hours for just this eventuality.* The section leader office hour schedule is available from the class web page. Please remember that you can talk to any section leader during office hours, not just the section leader running your weekly section.

On the due date, at the start of class, staple this page to the front of your neatly written (or, better yet, electronically-produced) answers and turn them in to the section leaders. Solutions submitted after the first five minutes of class on the due date will not be accepted. Want to be safe and submit your homework early? Feel free to do so.

— Points —			
Section	Exercises	Possible	Deductions
2.4	4(d)	2	
	10(a)	2	
	18(b)	2	
	36	6	
3.4	6	6	
	10(e)	2	
	18	2	
3.5	12(c)	2	
	20(a)	1	
	22(a)	1	
	26	2	
See Back	Program!	16	
	Subtotal:	44	
	Neatness:	1	
	Interactive Grading*:	5	
	Total:	50	
	Score:		

“Show” ≡ “Prove”
Ditto!

*If your score is below 35 of the 45 (44+1) points, to earn an additional five points you can arrange a time to meet with your section leader to review your mistakes. The meeting must be initiated within one week after the due date. If your score is 35 or better, you'll get those five points automatically.

Recursive “Proof” Generation. Our text provides an inductive proof of the Fundamental Theorem of Arithmetic (see page 285). The proof can be turned into a program that generates a “proof” (more accurately, a demonstration) for a given integer value. That is, the program shows, step by step, how the integer is decomposed into factors, and factors of factors, etc., until everything is shown to be products of primes.

Write a complete, well-documented Java program (named `Hmwk8.java`) that “proves” to the user that a given integer (provided on the command line) is either prime or can be shown to be expressible as the product of two or more primes. Here are two executions that show how your program’s output is to look:

```
$ java Hmwk8 1173
This program will demonstrate that 1173 is either prime
or is the product of two or more prime numbers.

1173 = 391 * 3; are these factors either prime or products of primes?
  391 = 23 * 17; are these factors either prime or products of primes?
    23 is prime.
    17 is prime.
  391 is the product of primes (23 and 17 are prime or prime products).
  3 is prime.
1173 is the product of primes (391 and 3 are prime or prime products).

As this output shows, the Fundamental Theorem of Arithmetic holds for 1173.
```

```
$ java Hmwk8 36
This program will demonstrate that 36 is either prime
or is the product of two or more prime numbers.

36 = 18 * 2; are these factors either prime or products of primes?
  18 = 9 * 2; are these factors either prime or products of primes?
    9 = 3 squared; is this factor either prime or product of primes?
      3 is prime.
    9 is the square of 3, which is prime or the product of primes.
    2 is prime.
  18 is the product of primes (9 and 2 are prime or prime products).
  2 is prime.
36 is the product of primes (18 and 2 are prime or prime products).

As this output shows, the Fundamental Theorem of Arithmetic holds for 36.
```

Note two features in particular: Indentation is used to make it easy to see what’s happening, and when a factor is shown to be the square of a prime the output reflects this.

Program Submission: After you have created your Java program, submit your completed program file(s) using the `turnin` command [on `lectura`](#). The submission directory is `cs245hw8`. Instructions are available from the brief `turnin` tutorial linked to the class web page. (The same tutorial explains one way to get files from your home PC to `lectura`, should you need to.) Remember to name your source code file `Hmwk8.java` so that we don’t have to guess which program to use with which of our tests.

Programming Info:

- Don’t know how to get information from the command line in Java? See `CmdLine.java`, which is linked to class web page.
- What will we be looking for in the way of program documentation?
See <http://www.cs.arizona.edu/people/mccann/style.html>; in particular, http://www.cs.arizona.edu/people/mccann/develop_java.html shows the basic content of the internal and external block comments you are to include with your code (along with a lot of stuff that isn’t really relevant to this class).

Yes, I want ‘a lot’ of documentation. Why? Because every programmer should include detailed, useful documentation in any program. The documentation is worth about 30% of your possible program score.