

CS102: Introduction to Computer Science

Summer 2014

Program #1

You are going to write a program that displays the prime factorization of a number specified by the user. The user should be prompted to enter a positive integer in the range of two to one million. If the user specifies an integers outside of this range, they should be re-prompted until they get it right. (You do not need to worry about them typing text or floating-point values.) The prime factors should be displayed separated by asterisks, with single spaces before and after each asterisk. Below are three sample runs (each run should display the result for a single integer).

```
Enter a positive integer from 2 to 1000000: -17
Enter a positive integer from 2 to 1000000: 2000000
Enter a positive integer from 2 to 1000000: 1000000
2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 5 * 5 * 5 * 5 * 5 * 5
```

```
Enter a positive integer from 2 to 1000000: 999983
999983
```

```
Enter a positive integer from 2 to 1000000: 714877
37 * 139 * 139
```

*Hints: Use a do...while loop to prompt the user for an appropriate integer. Then use a pair of nested loops to display the prime factorization. You do not need to do this in a particularly efficient manner. The outer loop can iterate from two to the specified number; these are the potential factors. You might think that you should use an "if" statement to check if each number is a factor; but since each number might be a factor several times, use a while loop (also making use of the modulus operator) instead. Whenever you find a factor, divide the original number by the factor (which can also decrease the number of iterations of the outer loop). To display the appropriate spaces and asterisks, there are two ways to go about that. Either display " * " after every factor except the last (you will know you found the final factor when the variable holding the original number becomes one); or display " * " before every factor except the first (you can use an extra variable that starts off as zero but gets set to one after the first factor is displayed).*

Your homework will be graded out of 100 points with the following breakdown:

- **Correctness:** You should follow all instructions exactly as stated above. This is an individual assignment (i.e., you should not collaborate with anyone else). **80 points.**
- **Elegance and Efficiency:** You should use the concepts we have learned in class to write your program in a simple, elegant manner. **10 points.**
- **Format:** Your program should use proper indentation and other spacing which makes the code readable and easy to understand. **5 points.**
- **Comments:** You should include one comment at the top of your program indicating your full name and (briefly) what the program does, plus one or more short comments within the code explaining how it works. **5 points.**

Submitting assignments: Email me your code (to CarlSable.Cooper@gmail.com) as an attachment. Do NOT attach the executable (only send the source file). Please state in your e-mail which environment you used to develop the code (e.g., Cygwin, Quincy, Ubuntu, etc.). This program is due the night of Tuesday, July 22, before midnight.