

Problem Solving using Python - Week 4 - Lab and Homework (10 pts = 100%)

Incremental Development, Test, Evaluation

General Idea

The goal of this lab and homework is to familiarize you further with the concept of black-box testing.

Intro

In the **homework** this week, you will be presented with three scenarios. In each, you will be given a specification for a program as well as access to a version of that program that has at least one bug. Your job is to test the program using whatever data seems most appropriate. Ultimately, you will summarize your work in a report to “management.”

You will need to test only one of the programs, which will given to you by the instructor.

In the **lab** you will see an an example.

From *programming problem solving model* point of view, you should preform only the *problem* and the *test* phases. Make sure that you understand the problem well before you start to test it. You don't want to test a solution to the wrong problem.

Lab (1 pts)

The first program concerns the *sum of the factors* of a given number. Open <http://blackbox.problemsolving.io> and choose **Sum of Factors** on the top menu.

Then, *after you think you found the bug*, check out the sample report written for this problem. Pay close attention to the contents. You will be writing three similar reports; you are not required to mimic the format exactly - indeed you are encouraged to individualize it for both the given problems and your personality, but you are expected to cover similar material.

We are aware that you can “cheat” on this and read the report first. Doing so will help you complete the lab hand-in, but will make homework more difficult.

Hand in

Hand in a single sheet with your name and a description of the bug. This is a description only - no need for a full report. A single sentence may simply state:

The program fails when the input is _____.

Important Note!

Each one of the programs have few **variants** with different bugs. The variants can be chosen in the website and we will use only the default variant (the ones that are set when you load the page). To prevent mistakes, we will also explicitly mentioned them in each one of the tasks. **Make sure you are working on the current variants!**

The screenshot shows the 'Black-Box Testing' website. At the top, there are four tabs: 'Sum of Factors' (selected), 'Simplified Sorting', 'Zulu Time', and 'Children's Games'. Below the tabs, the title 'Sum of Factors' is displayed, followed by an 'Overview' section. The overview text states: 'Many computer security protocols are based upon number-theoretical properties, particularly those related to factoring. Indeed, a good understanding of number theory is essential for any professional working with or designing such protocols. In this problem, we explore one of the first number-theoretical functions encountered by students studying in the field.' Below this is the 'Problem' section, which defines the sigma function $\sigma(n)$ as the sum of the positive divisors of n , where n is a natural number. It asks the user to compute the value of $\sigma(n)$ for a given value of n . A sample screenshot is shown below. The 'Input' section states: 'The input text field will hold a single natural number N . N will be between 200 and 2,000,000 inclusive.' The 'Output' section states: 'The output will consist of a single number expressing the value of $\sigma(n)$.' The 'Program' section contains a form with a 'Number' input field (placeholder: 'Enter number'), a 'Result' label, and a 'Variant' dropdown menu (currently showing '1'). Below the dropdown are 'Compute' and 'Clear' buttons. At the bottom of the page, it says: 'Originally written by David Levine. Re-implemented in web interface by Shlomi Hod.'

Figure 1: Variant

Homework (9 pts)

Instruction

You are to duplicate the work of the lab on each of the three programs. The catch is that this time you will have to write the report on your own and therefore cannot peak at the bugs before you begin.

1. Simplified Sort

The first program involves alphabetically (lexicographically) sorting three strings. If you are unfamiliar with the term, “alphabetical”, look it up on the internet.

Open <http://blackbox.problemsolving.io> and choose **Simplified Sort** on the top menu. Then perform the requisite analysis.

The Variant is 2.

2. Zulu Time

The second program involves time zones. Although we believe that the specifications are clear, feel free to look up “Zulu time” on the internet if you want more information. If you do so, however, remember that our problem description is “correct” no matter what the internet says.

Once again, open <http://blackbox.problemsolving.io> and choose **Zulu Time** on the top menu. Then perform the requisite analysis.

The Variant is 1.

3. Children’s Game

The third program involves a game that you have no doubt played many times.

For the last time today, open <http://blackbox.problemsolving.io> and choose **Children’s Game** on the top menu. Then perform the requisite analysis.

The Variant is 2.

Hand in

Hand in your report for the program you were given. Please write the report in **ONE** file (e.g., .md, .txt, .odt, .docx, .doc, .pdf)

Bonus (3 or 6 pts)

By submitting a report for one of the two additional programs, you may get up to 3 points extra. You can submit your report for one or two of the other problems.

Credit

Originally written by David Levine.