Computer Science I
Testing & Debugging

CSCI-141
Homework

09/19/2016

1 Problem

You are provided with code implementing two functions. The functions have not been tested, however. For each function, your tasks are to:

- Develop a broad test suite.
- Use a debugger to help you discover and fix any bugs in the code.
- Answer some related questions.

1.1 Task 1: Square root function

The provided file, `test_debug1.py`, contains a function `root()` that has one input parameter: a non-negative integer. The `root` function uses an algorithm to compute the integer portion of the square root of the input, without using the `math.sqrt()` function to do the computation.

The algorithm is described below. You can also look online for additional description (see `https://xlinux.nist.gov/dads/HTML/squareRoot.html`, for example).

We'll use the number 59368 as an example.

1. Starting from the ones digit and moving left, mark off pairs of digits. The leftmost 'pair' may only have one digit if the original number has an odd number of digits, but that's fine. We'll just treat it as having an extra 0 out front in that case. In this example we end up with:

$$05, 93, 68$$

2. We will loop through the pairs from left to right. Each iteration of the loop will ultimately generate one digit of the answer, in a process somewhat like long division. Putting the original number as the 'dividend', determine the largest integer (0-9) whose square is less than or equal to the first pair of digits in the original number. In this case the answer is 2, because $2^2 \leq 5$ and $3^2 \not\leq 5$. Put the 2 above the 'dividend' as part of the 'quotient', put the squared value below the first pair of digits, and subtract to get a 'remainder'.

$$\begin{array}{rr} & 2 \\ 2 & \sqrt{05, 93, 68} \\ & \underline{\ 4} \\ & 1 \end{array}$$

3. Similar to long division, bring down the next pair of digits. For the next iteration, the 'divisor' is double the current 'quotient' with one extra digit added to the end. What is the largest digit (0-9) that can be appended as a units digit to the current 'divisor' and then multiplied by itself such that the result is less than or equal to the current 'remainder' and brought down digits?

$$\begin{array}{rr} & 2\ ? \\ 2 & \sqrt{05,93,68} \\ & \underline{4} \\ 4\underline{?} & 1\ 93 \end{array}$$

Mathematically, for this example, what is the largest digit, d, such that $(40+d)\times d \leq 193$? The answer is 4, because $44 \times 4 \leq 193$ but $45 \times 5 \not\leq 193$. Thus 4 becomes the next digit in the 'quotient'.

4. We subtract to get a new remainder.

$$\begin{array}{rr} & 2\ 4 \\ 2 & \sqrt{05,93,68} \\ & \underline{4} \\ 4\underline{4} & 1\ 93 \\ & \underline{1\ 76} \\ & 17 \end{array}$$

5. Then bring down the next pair of digits and repeat from step 3.

$$\begin{array}{rr} & 2\ \ 4\ \ 3 \\ 2 & \sqrt{05,93,68} \\ & \underline{4} \\ 4\underline{4} & 1\ 93 \\ & \underline{1\ 76} \\ 48\underline{3} & 17\ 68 \\ & \underline{14\ 49} \\ & 3\ 19 \end{array}$$

The remainder at the end just means that the actual square root is somewhere between 243 and 244. We could continue the process with pairs of zero digits to the right of the decimal point, but we'll stop here and just calculate the integer position of the square root.

Also note that while the first 2 steps of the algorithm may look different than the repeated steps that follow, they really are the same, just with an initial 'quotient' that is 0 and no 'remainder' yet either.

1.2 Task 2: String splitting

The provided file, `test_debug2.py`, contains a function `alphaSplit` that has one input parameter: a string containing only lowercase English letters. The function uses an algorithm to determine whether or not it is possible to split the string into two substrings, satisfying the following constraints:

- Every character of the original string belongs to exactly one of the two substrings.
- The characters of each substring are in alphabetical order.
- Within each substring, the characters are in the same relative order that they appeared in the original string.

The final constraint just says that you can't shuffle the characters. Without this constraint, the problem is trivial because you can just sort the characters into alphabetical order.

For examples:

- "chbke" can be split into two such substrings: "chk" and "be".
- "gjba" can not be split into two such substrings. (If you put 'g' in substring 1 and put 'j' in substring 2, there is no valid place to put the 'b' or the 'a'. If you put both 'g' and then 'j' into substring 1 and then put 'b' in substring 2, there is no valid place to put the 'a'.)

Duplicate characters are allowed, and are considered to be in alphabetical order. For example, the string "fmcmh" can be split into "fmm" and "ch".

The function returns a boolean value. `False` if there is no valid way to split the string. `True` if there is at least one (there may be multiple) valid way to split the string.

2 Assignment Details

For each of the two tasks, you must do the following:

1. Download the appropriate code file that accompanies this document. **Follow the Homework 04 link on the course schedule page to find all materials for this assignment.**
2. Study the file's code content.
3. Develop a test suite containing pass/fail tests that cover different categories of inputs, and that uncover the fact that there are bug(s) in each provided program. Add a call to your test function at the bottom of the program so that when the program is run, it calls your test function.

4. Use a debugger and the knowledge from your tests to identify and fix any problems in the code. Take a screenshot of the debugger when you find an error (one screenshot per task is sufficient). You will include this with your submission. See section 2.1 on how to take screenshots in different OSs.
5. Update your test suite if necessary, and use it to validate the function.
6. Answer the following questions:
   (a) Give 2 examples of inputs for which the provided code gives a correct answer despite the fact that it is flawed.
   (b) For each example, explain why the faulty code produced the correct answer, despite the flaw(s).
   (c) Describe the bug(s) present in the code, and for each bug, indicate what the fix is.

2.1 How to take a screenshot

The task of taking a screenshot is different in the various operating systems. Below is a list of the operating systems and how to take a screenshot in each.

1. Windows
   In Windows there are two ways to take a screenshot.
   (a) Using the print screen button. When you are ready to take a screenshot, press the "Print Screen" (PrtSc) button on your keyboard. This will capture an image of your entire desktop. You must then open an image editing application, such as Paint, and paste the image into the canvas. Remember this is a screenshot of your entire desktop and may contain information you do not want to share. You can crop the image to show only the parts you want to share. Save the image, and you have your screenshot.
   (b) Using the Snipping Tool. This is easier than using the print screen button. Start the snipping tool application and select New. This will gray out your screen and give you a crosshair cursor. Click and hold the mouse button at the corner of the area you wish to capture, then drag to the opposite corner and release. This will capture the image in the Snipping Tool. Then save the image and you have your screenshot.
2. In Linux
   There are several ways to take a screenshot in Linux. Let's use Ubuntu as an example, but most of these will work on any Linux distro. If you aren't using GNOME, then GNOME-specific items won't work.
   (a) Using the print screen button. You can also take a screenshot of the entire screen by pushing the "Print Screen" (PrtSc) button on your keyboard. To get a screenshot of only the active window, use Alt-PrtSc. A dialog will appear asking you to save the image. Save the image and you have your screenshot. Be aware that the image may need some editing if you wish to not show everything contained in it.
   (b) Using gnome-screenshot. This can be done on the command line. Open a terminal and enter "gnome-screenshot". This will take a screenshot and open a

dialog to save it. Save it and you have your screenshot. Be aware that the image may need some editing if you wish to not show everything contained in it.

3. In OSX

   There are several ways in OSX to take a screenshot. I will only outline a few below.

   (a) Entire Screen. To take a screenshot of the entire screen press "Command + Shift + 3". This will place the screenshot image on your desktop in a .png format. Be aware that the image may need some editing if you wish to not show everything contained in it.

   (b) Part of the screen.

      i. Press Command+Shift+4. You'll see that your cursor changes to a crosshair pointer.

      ii. Move the crosshair pointer to where you want to start the screenshot.

      iii. Drag to select an area. To adjust the area, hold Shift, Option, or the Space bar while you drag.

      iv. When you've selected the area you want, release your mouse or trackpad button. Or to cancel, press Escape (esc).

      v. Find the screenshot as a .png file on your desktop.

3 Grading

Grading is described just with respect to Task 1. It totals 50%. The same grading applies to Task 2.

- 25% Task 1: updated code. Bug(s) have been fixed, and test function with ample pass/fail tests is included and called.
- 25% Task 1: writeup.
  - 5% for the screenshot.
  - 5% for 2 correct cases in which the function gives the correct answer despite being flawed (question (a)).
  - 5% for explanations of why these inputs yield a correct answer (question (b)).
  - 10% for discussion of the bugs and their fixes (question (c)).

4 Submission

Combine your screenshots and answers to the questions in a document. Convert your document to `pdf` format for submission.

Zip this document together (**.zip format)** with your updated code files (`test_debug1.py` and `test_debug2.py`).

Make sure your name is included in all submitted files.

5 points may be deducted for failure to follow these instructions.

Submit your zip file to the myCourses dropbox for this assignment.