

Assignment 2

This is an individual assignment worth 20% of the total unit score.

Due date: 11:00pm, Monday, 11 September 2023

DESCRIPTION

Program Under Test

Consider a program which splits an input list of integers into two lists of integers as output according to the following specifications:

- For the input list, the program is expected to accept a non-empty list of not more than 20 positive integers, where integer 0 is not regarded to be positive in this assignment. The input list may contain duplicated integers (that is, the same integer may appear more than once in the input list).
- For the output lists, one of them contains only odd integers in ascending order without duplication of integers, and the other list contains only even integers in ascending order without duplication of integers. Note that one of the output lists may be an empty list for some inputs.

For example, suppose that the input list is [3, 10, 9, 20, 16, 10, 9, 15, 7, 5, 28, 20, 5, 8, 20]. This list only contains positive integers and has 15 elements with some integers like 10 appearing more than once in the list. This is a valid test case. Its relevant outputs are two lists, one of which is [3, 5, 7, 9, 15] (only odd integers, in ascending order, no duplicated integers), and the other list is [8, 10, 16, 20, 28] (only even integers, in ascending order, no duplicated integers). Since the input is restricted to contain no more than 20 positive integers, the computed outputs can be easily validated in a manual approach, and hence you can assume that this program has no test oracle problem.

A program-under-test is given in either Python or Ruby in the Appendix.

Assignment Requirement

In this assignment, you are required to do the followings:

A. TASKS

Task 1: You are required to construct six valid concrete test cases for this program, which serve for different testing purposes or objectives. Valid test cases are non-empty lists of positive integers with not more than 20 elements, where some elements could be duplicated and integer 0 is not regarded as positive in this assignment.

Apart from explicitly describing your concrete test cases, you must describe and justify their purposes, and explicitly describe their relevant outputs. For example, one valid concrete test case can be [3, 10, 9, 20, 16, 10, 9, 15, 7, 5, 28, 20, 5, 8, 20]. Its output should be [3, 5, 7, 9, 15] and [8, 10, 16, 20, 28]. This test case contains duplicated integers and hence can serve to test the proper removal of duplicated integers in the output lists. Another example is the test case [13, 7, 9, 15]. Its outputs should be [7, 9, 13, 15] and the empty list denoted by []. This test case serves to test the scenario that the output list of even integers is empty.

Task 2: Suppose that due to testing resources, you cannot test the program with all your six test cases constructed above. You can only afford to *test your program with one and only one test case*. What will be your choice of this single test case among the 6 test cases proposed by you in Task 1? Explain and justify your choice.

Task 3: Conduct the testing with any one of two programs given in the Appendix with all your test cases in Task 1 and Task 2. Note that the program may contain bugs. Report the individual test cases and the results, as well as discuss the insights of test outcome including the suggestion to improve the program.

B. REQUIREMENTS

- **You need to prepare a self-contained and complete report for addressing these three tasks.** Your report must use single column, 12-point font size on A4 papers. Your report must contain no more than three pages. Coverage page is not required, but your submission must contain your full name and your student number on the first line of its first page.
- **You must submit the Assignment 2 report as a single PDF file** (which must have the filename specified in this format as “Assignment2-YourStudentID-YourSurname.pdf”, for example, Assignment2-12345678-Nash) before the due date using the Submission System for Assignment 2 in Canvas available to you.

C. INSTRUCTIONS

In “Lecture 3.pdf”, we discuss how to use the information about the inputs and outputs for a program to define test cases without referencing to the codes of the program under test. You are required to use this approach to complete the first two tasks. For Task 3, you must focus on testing the function `split_and_sort()` only.

MARKING CRITERIA

- o **A maximum of 40 marks** for Task 1 which include 5 marks for each test case description, its output, its justification and 10 marks for the quality and comprehensiveness of its description and presentation.
- o **A maximum of 20 marks** for the explanation and justification of your choice for Task 2.
- o **A maximum of 30 marks** for the test case description, the test execution and results, the insights of discussion as well the program improvement for Task 3.
- o **A maximum of 10 marks** for the presentation, structure, comprehensibility, completeness and cohesion of report, as well as the compliance with the submission requirements.
- o **Penalty** will be applied for late submission and plagiarism. Refer to the Unit Outline for the policy on late submission and plagiarism.

APPENDIX

You can use either Python or Ruby program below for your Task 3. It can be downloaded in Canvas as well.

Program under test in Python **assignment2.py**.

```
def split_and_sort(nums):
    # check if input list length is less than or equal to 20
    if len(nums) > 20:
        return "Error: Input list should not contain more than 20 integers."

    # check if 0 is in the input list
    if 0 in nums:
        return "Error: The number 0 is not a valid input."

    # filter odd and even numbers into two separate lists
    odd_nums = [num for num in nums if num % 2 == 1]
    even_nums = [num for num in nums if num % 2 == 0]

    # remove duplicates and sort
    odd_nums = sorted(odd_nums)
    even_nums = sorted(even_nums)

    return odd_nums, even_nums

nums = [5, 4, 6, 10]
odd_nums, even_nums = split_and_sort(nums)

print("Odd numbers:", odd_nums)
print("Even numbers:", even_nums)
```

Program under test in Ruby **assignment2.rb**.

```
def split_and_sort(nums)
  raise "Error: Input list should not contain more than 20 integers." if nums.length > 20
  raise "Error: The number 0 is not a valid input." if nums.include?(0)

  odd_nums = nums.select { |num| num.odd? }.sort
  even_nums = nums.select { |num| num.even? }.sort

  [odd_nums, even_nums]
end

nums = [5, 4, 6, 10]
odd_nums, even_nums = split_and_sort(nums)
puts "Odd numbers: #{odd_nums}"
puts "Even numbers: #{even_nums}"
```