Computer Programming
Spring 2020

CSI2100-01 Lab 7

Bernd Burgstaller
Yonsei University

Overview

- Questions
- Programming Problems
- Deliverables, Due Date and Submission
- Notes:

  1) Please consider using the OnlinePythonTutor when debugging your code.
  2) Some lab problems can be tested with Hyeongjae Python.
     - Please refer to the last page of this assignment for details.
  3) Because of our coronavirus precautions, you are provided with an excerpt of the relevant sections of the textbook together with this lab assignment.

Questions

You are kindly asked to submit the answers to the questions on the following page in a file named **README.txt**

Question 1: Textbook page 202, Exercise 1.

Question 2: Textbook page 203, Exercise 6.

Question 3: What is the output of the below program? Justify your answer. PythonTutor may be helpful to show what is happening.

```
l1 = [1,2]
l2 = [0,5]
l3 = ['foo']
l1.append(l3)
l2[0] = l3
print(id(l2[0]) == id(l1[2]))
```

Programming Problems

Note: For Problems 1 — 3, you should **only** hand in the code of your Python function, **nothing else!**

- No global variables, no main program to call the function, no etc!

Problem 1: Textbook page 203, Exercise P1.

The textbook states 'true' and 'false', meaning '`True`' and '`False`' as Python Boolean literals. Your function should accept 3 integer arguments. Therefore your function needs 3 formal parameters.

Problem 2: Textbook page 203, Exercise P2.

- Your function should accept 3 integer arguments. Therefore your function needs 3 formal parameters.
- We use <= for the ordering relation. Example: `ordered3(1,1,1)` shall return `True`.

Problem 3: Exercise P3 from page 203 of the textbook.

- The numbers 1 and n are included in the interval.
- Your program **can assume** that m <= n. This is a so-called **precondition** for your function. If the caller of your function does not obey this precondition, your function is allowed to fail. For example, the following call would violate the precondition: `modCount(100, 1000)`.
  Preconditions are contracts between the user and the implementer of a function. This is important with large projects where the user is different from the implementer.

Problem 4: Write a function that evaluates the polynomial

$$3x^5 + 2x^4 - 5x^3 - x^2 + 7x - 6$$

Your program should ask the user to enter a value for x, call the function to compute the value of the polynomial, and then display the value returned by the function.
Example:

```
Enter a value for x: 5
Polynomial for x=5: 10004
```

Hint 1: you can assume that the user input is of type integer.

Hint 2: your function **must start** with the following line (otherwise automated grading will fail, resulting in 0 points):

```
def evalPolynomial(x):
```

Problem 5: Books are identified by an International Standard Book Number (ISBN). ISBNs assigned after January 1, 2007 contain 13 digits, arranged in five groups. For example, 978-0-470-55515-6 is a valid ISBN. The first group of digits (the GS1 prefix), is currently either 978 or 979. The group identifier specifies the language or country of origin.

(continued on next page)

The publisher code identifies the publisher (470 is the publisher code of Wiley). The item number is assigned by the publisher to identify a specific book (55515 is the code for our textbook). An ISBN ends with a check digit that's used as a checksum for the preceding digits.

Your task is to write a program that breaks down an ISBN entered by the user.

Example:

```
Enter an ISBN: 978-0-470-55515-6
978.................GS1 prefix
0...................Group identifier
470.................Publisher code
55515...............Item number
6...................Check digit
```

Note: The number of digits in each group may vary. You're not allowed to assume that the groups always have the lengths shown in this example. You can test your program with real ISBN numbers found on the back-cover and on the copyright page of a book.

Hint: the main problem to overcome in this case is to split the user-provided input string into the 5 digit groups. To our help, the digit groups are separated by a dash. We can use this fact with the **split** method. If **isbn** is a string containing a valid ISBN number, then **isbn.split('-')** allows us to retrieve a list of substrings, split across the occurrences of the dash character. The ISBN from the previous example would be split into the list ['978', '0', '470', '55515', '6'].

Note 2: the field-width for each digit group plus the dots (...) is 20 characters. This way the digit group descriptions are vertically aligned at their left-hand side (see the example on the previous page). Your program must format the output correctly for arbitrary ISBNs. For example, your program could receive a two-digit publisher code, but still the field-width stays as 20 characters.

In case you concern about overly long digit groups: as stated previously, ISBN numbers issued since 2007 (what we're using for this problem) consist of 13 digits. ☺

Marking Criteria and Plagiarism

- Marking Criteria
  - Score is only given to programs that compile and produce the correct output with Python version 3.
  - Points are deducted for programs that are named wrongly. See the list of deliverables for the required file names.
  - Points are deducted for programs that produce warnings.
  - Points deductions on programming style: please provide comments in your code.
    - Please provide docstrings with **all** functions.
  - Please pay particular attention to the **requested output format** of your programs. Deviating from the requested output format results in points deductions.

- Plagiarism (Cheating)
  - This is an individual assignment. All submissions are checked for plagiarism.
  - Once detected, measures will be taken for **all** students involved in the plagiarism incident (including the ``source'' of the plagiarized code).

Deliverables, Due Date and Submission

- Please prepare the files for the programming problems and questions. The names of the files, their due dates and the archive file-name is given in the below table.
  - Please upload your archive file by the stated due date on YSCEC.
- Lab problems marked as '✓' can be tested on our Hyeongjae Python site http://hyeongjaepython.elc.cs.yonsei.ac.kr/

Problem	File name	Due	Archive name	Hyeongjae Python
1	lab7_p1.py	Wednesday May 6, 2020, 23:00	lab7_<student id>.zip	✓
2	lab7_p2.py			—
3	lab7_p3.py			—
4	lab7_p4.py			—
5	lab7_p5.py			✓
Questions	README.txt			—