CSE 231 **Fall 2013**

Computer Project #3

Assignment Overview

This assignment focuses on the design, implementation and testing of a Python program which uses control structures and strings to solve the problem described below.

It is worth 30 points (3% of course grade) and must be completed no later than 11:59 PM on Monday, September 23.

Assignment Specifications

Some whole numbers can be split into smaller pieces, where each piece contains the same number of digits. For a subset of the whole numbers which can be split evenly, the pieces are in numerically increasing order. Examples:

- 154152 can be split evenly into two-digit pieces (15, 41, 52) – increasing order
- 154152 can be split evenly into three-digit pieces (154, 152) – not increasing order
- 154152 cannot be split evenly into five-digit pieces
- 173 can be split evenly into one-digit pieces (1, 7, 3) – not increasing order
- 173 can be split evenly into three-digit pieces (173) – increasing order
- 173 cannot be split evenly into two-digit pieces

Design, implement and test a Python program which checks to see if a user-supplied whole number can be split evenly into pieces which are in numerically increasing order.

1. The program will prompt the user to enter a whole number. If the user enters an invalid input (anything other than a whole number), the program will repeatedly prompt the user until a valid input is entered.

2. The program will then prompt the user to enter the number of digits in each piece. The program will verify that the input is valid (a whole number which is a proper divisor of the number of digits in the first input); if the input is invalid, the program will repeatedly prompt the user until a valid input is entered.

3. The program will split the number into pieces and display those pieces on one line (separated by commas).

4. The program will report whether or not the pieces are in numerically increasing order. If there is only one piece, it is defined to be in numerically increasing order.

Assignment Deliverables

The deliverable for this assignment is the following file:

proj03.py – the source code for your Python program

Be sure to use the specified file name ("proj03.py") and to submit it for grading via the **handin** system before the project deadline.

Assignment Notes

1. To clarify the project specifications, we've provided sample output at the end of this document.

2. You may not use any collection (such as a list, tuple, or dictionary) in your program.

3. Be sure to prompt the user for the two inputs in the correct order. Also, your program cannot prompt the user for any other inputs. To grade your program, your TA will enter a series of inputs.

4. You may find it easier to process the large whole number as a string when you split it into pieces.

One approach:

```
start with an empty string to hold the substring, i.e. the split
as you loop through the whole number one digit at a time
    append digits to the substring
    if the substring reaches the desired length
        do something with the substring
        reset the substring to empty to collect the next substring
        reset the substring length count to zero
```

An alternative approach:

```
figure out how many substrings you want
loop that number of times
    use slices to create the substrings
```

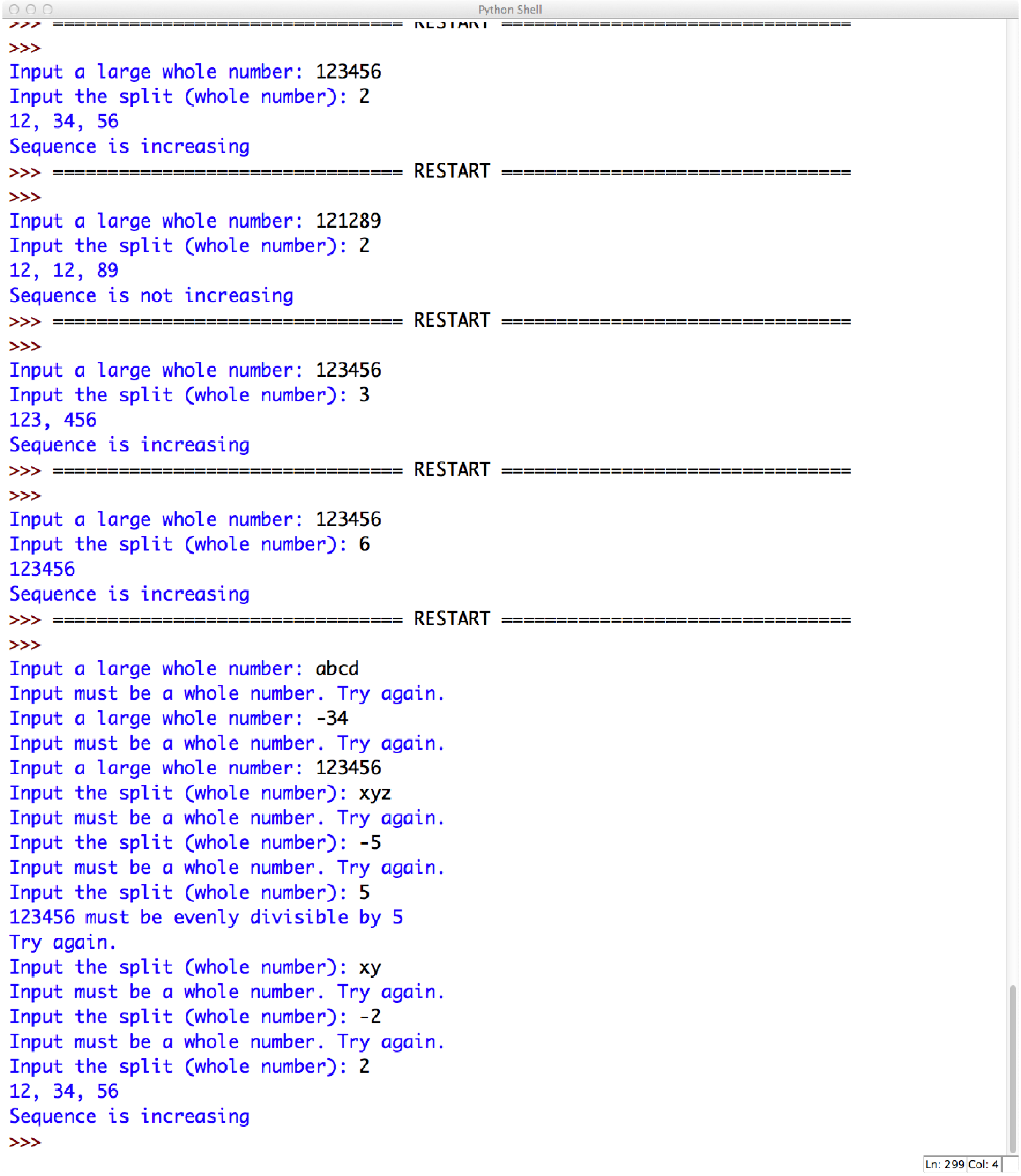
5. You can use the `isdigit()` string method to determine if a string contains only digits. Type `help(str.isdigit)` in the Python shell for more information.

6. You can use the `len()` function to determine the length of a string. Type `help(len)` in the Python shell for more information.

7. Getting the commas right in the output is surprisingly tricky because you only want commas *between* numbers, so leave that until everything else is working correctly.

8. Leave error checking of the input until last. Begin by building your program assuming perfect input. However, plan on quite a bit of time to get the input checking details right.

Sample Output



```
>>> ================================ RESTART ================================
>>>
Input a large whole number: 123456
Input the split (whole number): 2
12, 34, 56
Sequence is increasing
>>> ================================ RESTART ================================
>>>
Input a large whole number: 121289
Input the split (whole number): 2
12, 12, 89
Sequence is not increasing
>>> ================================ RESTART ================================
>>>
Input a large whole number: 123456
Input the split (whole number): 3
123, 456
Sequence is increasing
>>> ================================ RESTART ================================
>>>
Input a large whole number: 123456
Input the split (whole number): 6
123456
Sequence is increasing
>>> ================================ RESTART ================================
>>>
Input a large whole number: abcd
Input must be a whole number. Try again.
Input a large whole number: -34
Input must be a whole number. Try again.
Input a large whole number: 123456
Input the split (whole number): xyz
Input must be a whole number. Try again.
Input the split (whole number): -5
Input must be a whole number. Try again.
Input the split (whole number): 5
123456 must be evenly divisible by 5
Try again.
Input the split (whole number): xy
Input must be a whole number. Try again.
Input the split (whole number): -2
Input must be a whole number. Try again.
Input the split (whole number): 2
12, 34, 56
Sequence is increasing
>>>
```