COMP 10280
Programming I (Conversion)

Practical Sheet 14
Thursday, 10 October 2019

For each of the following questions, write an algorithm in pseudocode first before writing a Python program. Submit your algorithms in pseudocode as well as your Python programs.
When writing functions, use one-line or multi-line docstrings, as appropriate, to document your functions.

1. Take the two programs that you have previously written to implement the recursive and the non-recursive definitions, respectively, of the factorial function. Supplying increasingly large inputs to the program, note the times taken for each program to calculate the factorial of an integer.

   Report your findings in `p14q1.txt`.

2. Again, taking the two programs that you have previously written to implement the recursive and the non-recursive definitions, respectively, of the factorial function, progressively increase the size of the number that is entered to the program until your program crashes or reports an error.

   Report your findings, in particular the difference in behaviour between the two programs, in `p14q2.txt`.

3. Implement the program that searches for prime numbers in a range of integers and demonstrates the use of the `else` clause in a `for` loop in Python from recent lectures (Lecture 13) (Page 16 of the notes). Ensure that you understand how this program works.

   Save this program as `p14p3.py`.

4. Using the program developed in Question 3 as a starting point, or using a completely new technique if you prefer, write a program that searches for prime numbers in a range of integers. Again, the program should print out an appropriate message if a number is a prime number. However, instead of printing out the first pair of factors that it discovers for a non-prime number, this program should print out *all* the pairs of factors.

   Save this program as `p14p4.py`.

5. The Fibonacci Series may be defined formally as follows:

$$f(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ f(n-1) + f(n-2) & n > 1 \end{cases}$$

   (a) Write a recursive function that takes as its single argument a positive integer and prints out that number of terms from the Fibonacci Series.

   (b) Write a program that prompts the user for a series of integers. For each number entered the program should check that it is non-negative. If it is, it calls the function defined in part (a). The program should stop when a non-positive number is entered.

   (c) In your function, include some print statements that allow you to see the operation of the recursion and its progress towards the base cases.

   Save this program as `p14p5.py`.

Please upload your work to the Brightspace site before Monday evening.

You should keep a copy of your programs for your portfolio.