Theory and Practice of Programming
Homework Sheet 1
Due: Friday 24th October 12 noon
Write your name and your group (i.e. Group 4C) at the top of your answer sheet
Find your group on the TPOP theory homework webpage

Try all questions and submit your written answers to CS reception. Any questions you cannot complete can be discussed in problem class. Working in groups is encouraged but you must submit your own answers. You should include a print out of your python code where appropriate. Email william.smith@york.ac.uk if anything is unclear.

1. [Easy] Think about the 'performance' of a computer program. The most obvious measure of efficiency is speed, or the time taken for the program to execute. What other measures of efficiency might one use in a real-world setting?

2. [Easy] Thinking about computer software in general, are there other factors which are more important than performance? Try to rank them in order of importance.

3. [Straightforward] We now go all the way back to 300BC and Euclid's algorithm for finding the greatest common divisor of two integers.

   Input: two positive integers, a and b where a is the larger of the two
   Algorithm:

   (i) Divide a by b and store the remainder in r
   (ii) If r equals 0 then stop: the greatest common devisor is b
   (iii) Otherwise, copy b into a and r into b and return to step (i)

   "Tracing" an algorithm involves producing a "trace table" which has a column for each variable and a row for each executed line of your algorithm. The first column should show the current line number and the first row should show initial values (unknown values are labelled "?"). You "run" the algorithm by hand by adding a row after each step of the algorithm, updating the contents of variables as you go. See: youtu.be/b5Ts7f3Sp4A for a good explanation. You should include a column for boolean values resulting from comparisons. Trace Euclid's algorithm when called with the following values:

   (a) $a = 3$, $b = 1$
   (b) $a = 36$, $b = 15$
   (c) $a = 1071$, $b = 462$

4. [Medium] In the algorithm above, we assumed we had a means to perform division and compute remainders. Imagine implementing Euclid's algorithm on a computer which only supports the mathematical operations of addition and subtraction.

   Devise an algorithm for computing the quotient, q, and remainder, r, when dividing an integer a by an integer b **without using division**. In other words, given a and b, find integer values for q and r so that $a = qb + r$ and $r < b$.

5. [Straightforward] Trace the values of each variable through each step of your division algorithm when called with the same pairs of values as in question 3.

6. [Implementation] Implement the algorithm used in question 3 and use it to find the greatest common divisors of 12,345 and 987,654,312. The pseudocode includes an instruction "return to step (i)" which is not something we can do in Python. Think about how to express the algorithm using a while loop instead.

7. [Implementation] Now implement division using the algorithm designed in question 4 and check that you get the same answer. Is your division function noticeably slower than the standard Python division operator?

8. [Challenge] You are writing a program for an archaic computer which only has enough memory to hold two integer variables! You need to swap the contents of the two variables. If you had a third temporary slot, this would be straightforward (e.g. to swap variables a and b using c as a temporary slot you do: $c := a$; $a := b$; $b := c$). Note, I use the symbol := for assignment (i.e. putting a value into a variable) and = to test equality. This is consistent with the normal meaning of the equals symbol but is different to Python which uses = for assignment and == for testing equality.

   Show how to swap variables a and b without an additional variable using only addition and subtraction.