

22C:16 Homework 2

Due via ICON on Wednesday, Feb 8, 4:59 pm

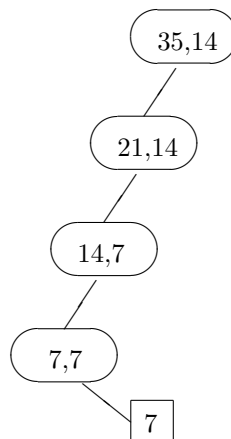
What to submit: Your submission for this homework will consist of three files. One of them will be a pdf file called `homework2.pdf`. This will contain the answers to Problems 1, 2, 3, 5, 6, and 8. This pdf file should start with your name, section number and student ID. The second file should be called `GCD1.py`. This is a python program file containing the answer to Problem 4. This file should also start with your name, section number and student ID appearing at the top of the file as Python comments. The third file should be called `GCD2.py`. This is a python program file containing the answer to Problem 7. It should start with your name, section number and student ID appearing at the top of the file as Python comments. You will get no credit for this homework if your files are named differently, have a different format (e.g., docx), or if your files are missing your information.

Euclid's algorithm for the greatest common divisor. The greatest common divisor (GCD) of two numbers is the largest number that divides evenly (i.e., without a remainder) into both numbers. For example, the GCD of 12 and 16 is 4.

1. Euclid's algorithm computes the GCD of two numbers. It is based on the principle that the GCD of two numbers does not change when the smaller number is subtracted from the larger number. For example, the GCD of 35 and 14 is also the GCD of 21 and 14. Explain briefly why this is true.

Hint: Suppose that d is the greatest common divisor of the two input numbers. Then the larger number can be written as $a \times d$ for some positive integer a and the smaller number can be written as $b \times d$ for some positive integer b . Now consider the expression for the larger number minus the smaller number.

2. The property described in problem 1 can be applied repeatedly until the two numbers are equal. This number is the GCD of the original two numbers. An execution tree of this algorithm for 35 and 14 is shown below. Construct similar trees to find the GCD of 96 and 36 and the GCD of 63 and 27.



3. Write the pseudo code for the algorithm described in problem 2.

Hint: If you need help getting started, look at the pseudo code for problem 6.

4. Turn the pseudo code of problem 3 into a Python program that reads in two non-negative integers and produces their GCD. Type this program into a Python IDE and save it as a file called GCD1.py. Your program should include comments as discussed in the class lecture, including a section at the start that includes your full name, section, and student ID. Your program may assume that the user will input only non-negative integers and does not have to do any error checking, but it should not require the larger number to be entered first.
5. Test your program by turning your birthdate into a decimal number. For example, if you were born on June 9th, 1995 you would use the number 19950609. To answer this question, write down your birthdate and the GCD of it and the numbers 75, 144, and 390.
6. The pseudo code for a more efficient implementation of Euclid's algorithm is given below:

```
Enter the larger number (the dividend)
Enter the smaller number (the divisor)
Calculate the remainder of dividing the dividend by the divisor
While the remainder is greater than zero
    Set the dividend to the value of the divisor
    Set the divisor to the value of the remainder
    Calculate the remainder of dividing the new dividend by the new divisor
Output the remainder as the GCD
```

Describe briefly the principle (similar to the one described in problem 2) on which this algorithm is based.

7. Turn the pseudo code of problem 6 into a Python program that reads in two non-negative integers and produces their GCD. Type this program into a Python IDE and save it as a file called GCD2.py. Your program should include appropriate comments as described in problem 4. Your program may assume that the user will input only non-negative integers and does not have to do any error checking. Test your program on the same numbers you used in problem 5 to see if it returns the same GCDs.
 8. Use your program to compute the GCD of February 8, 2012 and the numbers 75, 144, and 390.
-