Programming Design, Spring 2013
Homework 04

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

To submit your work, please upload the following two files to the online grading system PDOGS at http://stella.im.ntu.edu.tw/online-judgement/.

1. Your source file (the .cpp file) for Problem 1. This should be submitted to the PD002 section.
2. Your source file (the .cpp file) for Problem 2. This should be submitted to the PD003 section.

NO hard copy and NO late submission. The due time of this homework is ***1:00pm, March 18, 2013***.

Problem 1

(40 points) Please write a C++ program according to the following instructions.

What should your program do

The input contains several lines of numbers. In each line, there are at most 101 numbers with the last number being -1. All other numbers are nonnegative and may be fractional. The number of numbers is a line must be odd. Therefore, by excluding the last number -1, we may pair the remaining nonnegative numbers.

Your program should process these nonnegative numbers as follows. Suppose there are $2n+1$ numbers, including -1 in a line. Let these numbers be $x_1, x_2, ..., x_{2n}$, and -1. The program should output

$$\max_{i=1,\ldots,n} \Big\{ x_i + x_{2n-i+1} \Big\}.$$

In other words, it should pair the first number with the $2n$th number, the second number with the $(2n-1)$th number, ..., and find which pair generates the largest sum.

When the program output the result, the sum should be printed out with two digits after the decimal point. Remaining digits should be truncated in any case. After this number, a new line character should be placed.

Below are some examples:

- Input: `2 3 15.275 8 -1`. Output: `18.27`. Please note that the output should not be `18.28`.
- Input: `1 2 3 4 5 6 -1`. Output: `7.00`. It does not matter which pair generates this sum.
- Input: `9.246 6.125 32.456 87.325 16.6 32.65 -1`. Output: `119.78`.

Grading criteria

- 70% of your grades for this program will be based on the correctness of your output. The online grading system will input 35 sets of testing data and then check your outputs. You may only see the grades of running your program on these data but cannot see the inputs and outputs. These 35 sets count for 70 points, i.e., 2 points for each set.
- 30% of your grades for this program will be based on how you write your program, including the logic and format. Please try to write a robust, efficient, and easy-to-read program.

Problem 2

(60 points) Please write a C++ program according to the following instructions.

What should your program do

The input contains several lines, each with two positive integers no greater than 100000. These two positive integers are separated by one space. Your program should find the greatest common divisor of these two positive integers by running the Euclidean algorithm and output the solution process.[1]

Let the two integers be p and q. Your output should be formatted in the following way. First, you output these two numbers, first the larger one, then a space, then the smaller one, then a colon, then a space. After these, you need to output all the remainders you go through when running the algorithm until you output the greatest common divisor (which may be 1). These numbers should all be separated with white spaces, and after the greatest common divisor there should be a period. Between the greatest common divisor and the period, there should be no space. As usual, at the end of this line of output, there should be a new line character.

Below are some examples:

- Input: `952 1387`. Output:

  `952: 435 82 25 7 4 3 1.`

- Input: `124 48`. Output:

  `124 48: 28 20 8 4.`

- Input: `3 100`. Output:

  `100 3: 1.`

Grading criteria

- 70% of your grades for this program will be based on the correctness of your output. The online grading system will input 35 sets of testing data and then check your outputs. You may only see the grades of running your program on these data but cannot see the inputs and outputs. These 35 sets count for 70 points, i.e., 2 points for each set.
- 30% of your grades for this program will be based on how you write your program, including the logic and format. Please try to write a robust, efficient, and easy-to-read program.

[1] The description of the Euclidean algorithm can be easily found online, e.g., on Wikipedia. It is highly possible that you have learned it in your elementary school time.