# Java Development Homework

Due before 2024 March 20 9:00am

# 注意事項

1. 開始作答之前，先確定自己是否已熟悉 online judge (自動評測系統) 的操作流程。
2. Online judge會在截止日期之後，評測同學所交到Moodle的程式碼，寫完程式後，請務必繳交到Moodle。
3. Online judge帶有程式相似度比對系統，抄襲程式者與被抄襲者將一律0分。
4. 對於題目有任何問題，請聯繫助教。

# Homework 1

# Problem Description

**Problem:** Big Floating Point Number Addition

**Description:**
In this problem, you are tasked with implementing a method to accurately add two large floating-point numbers represented as strings. These numbers may be too large or too precise to be handled by standard Java floating-point data types like float or double. Your function must take two strings representing non-negative floating-point numbers and return their sum as a string.

**Notes:**

1. Can assume the inputs will be non-negative

# Problem Description (中文)

**Problem:** 大浮點數運算

**Description:**
請實現一個方法，以準確地加上兩個用字串表示的大浮點數。這些數字可能太大或太精確，無法由標準Java浮點數類型（如float或double）處理。輸入必須接受兩個代表浮點數的String，輸出為兩個數字加起來之後的結果。

**Notes:**

1. 可以假設輸入為非負的浮點數。

# Sample Input and Output

| | |
|---|---|
| Keyboard Input | 123.45 987.65<br>456.789123 9876.54321<br>0.0 0.0<br>99999.9999999999 0.0000000001 |
| Output | 1111.10<br>10333.332333<br>0.0<br>100000.0000000000 |

**Explanation:**
For the first input “123.45” and “987.65” separated with a space, you should return and print the result after adding both numbers, in first example it’s “1111.10”

**Notes:**

- Please use java.util.Scanner for handling input

# Sample Input and Output (中文)

| | |
|---|---|
| Keyboard Input | 123.45 987.65<br>456.789123 9876.54321<br>0.0 0.0<br>99999.9999999999 0.0000000001 |
| Output | 1111.10<br>10333.332333<br>0.0<br>100000.0000000000 |

**Explanation:**
以第一行輸入為例，“123.45”及“987.65” 中間以空格隔開，輸入完之後會進行加法運算，並印出答案 “1111.10”

**Notes:**

- 請使用java.util.Scanner來處理輸入

# Submission

Please archive your source code to STUDENT_ID.zip (download the example zip file from Moodle) and **upload to Moodle Homework 1** before deadline.

Your zip file should follow the following format.

STUDENT_ID.zip

|- src

|- META-INF

| |- MANIFEST.MF

All the source files (*.java) are put in the src directory.

The entry point (i.e. main class) of the program is specified in the MANIFEST.MF file.

No late submission is accepted.

# Homework 2

# Problem Description

A pure infinitely-repeating decimal number is defined as a decimal where all digits in the fractional part are non-zero and repeat infinitely. For example, $3.\overline{142857}$ represents a pure infinitely-repeating decimal number with the repeating sequence 142857. Conversely, $3.6\overline{142857}$ is not a pure infinitely-repeating decimal number as it contains non-repeating digits in its fractional part. In this problem, it is assumed that all decimal numbers provided are pure infinitely-repeating. Please write a program to compute the sum of two pure infinitely-repeating decimal numbers.

**Input Format (Please use java.util.Scanner to read the input.)**

The first line is an integer n indicating the number of test cases. Each test case consists of two pure infinitely-repeating decimal numbers x and y. Both x and y may be positive. For example, 9.2 indicates $9.\overline{2}$; 3.56 indicates $3.\overline{56}$.

**Output Format**

For each test case consisting of pure infinitely-repeating decimal numbers x and y, print the sum of x and y.

Note that your output may either a pure infinitely-repeating decimal number or an integer. Even though $0.14\overline{285714}$ and $0.\overline{142857}$ are numerically equal, the former is not allowed. For $0.\overline{9}$, your program should output an integer, rather than a decimal.

Each test case consists of two decimal numbers x and y from (0 , 100)

# Problem Description (中文)

在一個小數點中，其小數的部分非零且為無限循環重複的小數，被稱為純無限循環小數。舉例來說， $3.\overline{142857}$是一個純無限循環小數，其循環部分為142857；而$3.6\overline{142857}$不是純無限循環小數，因為它的小數部分包含非循環的數。在此題目中，所有的小數皆為純無限循環小數，請寫一個程式來計算兩個純無限循環小數之和。

**輸入格式** (請使用java.util.Scanner讀取輸入)

第一行是一個整數 n，表示測試案例的數量。第二行開始是測試案例，每個測試案例包含兩個純無限循環小數 x 和 y ，中間以空格隔開。x 和 y 皆為正數，例如，在測資中，9.2 表示 $9.\overline{2}$ ；3.56 表示 $3.\overline{56}$ 。

**輸出格式**

對於每個由純無限循環小數 x 和 y 組成的測試案例，請輸出 x 和 y 的和。

**請注意**

1. 在輸出時可能是一個純無限循環小數或一個整數。
2. 雖然$0.14\overline{285714}$和 $0.\overline{142857}$在數值上相等，但前者不被允許，答案輸出應為0.142857。
3. 當合為 $0.\overline{9}$ ，程式應該輸出整數1，而非小數1.0。
4. x與y的範圍為 0 < x < 101 , 0 < y < 100

# Sample Input and Output

| | |
|---|---|
| Keyboard Input | 2<br>9.2 3.56<br>2.34 1.234 |
| Output | 12.78<br>3.577668 |

**輸出說明**

當兩個連續循環小數，循環位元不同時需要疊加循環到相同的小數點位數

(1) 9.2 + 3.56

9.22 + 3.56 =12.78

(2) 2.34 + 1.234

2.343434 + 1.234234 = 3.577668

| | |
|---|---|
| Keyboard Input | 2<br>9.876 1.2345<br>6.8 0.142857 |
| Output | 11.111400329222<br>7.031746 |

**輸出說明**

當連續循環小數的循環節首位相加需進位時，則循環節的末位也必須進位

(3) 6.8 0.142857

6.888888 + 0.142857 = 7.031756 (末位進位)

# Sample Input and Output

| | |
|---|---|
| Keyboard Input | 2<br>1.234 4.321<br>88.3 11.6 |
| Output | 5.5<br>100 |

**輸出說明**

小數部分只需顯示循環的數字

當合為 $0.\overline{9}$ ，程式應輸出整數

(4) 1.234 + 4.321 = 5.5 (而非輸出5.555)

(5) 88.3 + 11.6 = 100 (而非輸出99.9)

**API提示:**

- double java.lang.Double.parseDouble(String s)
- String java.lang.Double.toString(double d)
- String java.lang.String.substring(int beginIndex)
- double java.lang.Math.pow(double a, double b)
- long java.lang.Math.round(double a)
- String java.lang.String.format(String format, Object... args)

# Submission

Please archive your source code to STUDENT_ID.zip (download the example zip file from Moodle) and **upload to Moodle Homework 2** before deadline.

Your zip file should follow the following format.

STUDENT_ID.zip

|- src

|- META-INF

| |- MANIFEST.MF

All the source files (*.java) are put in the src directory.

The entry point (i.e. main class) of the program is specified in the MANIFEST.MF file.

No late submission is accepted.