

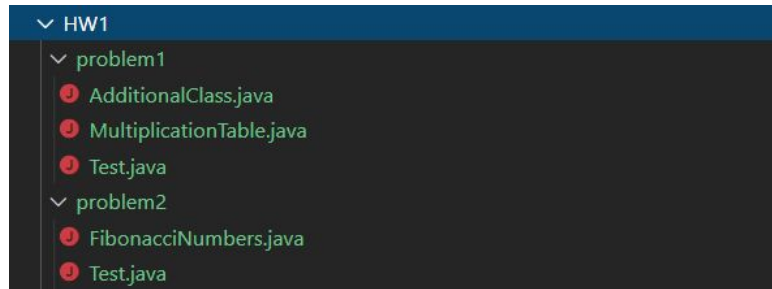
AY 2020 Assignment 1 [5 Marks]

Date / Time	11 September 2020 – 25 September 2020 23:59
Course	[M1522.600] Computer Programming
Instructor	Youngki Lee

- You can refer to the Internet or other materials to solve the assignment, but you ***SHOULD NOT*** discuss the question with anyone else and need to code **ALONE**.
- We will use the automated copy detector to check for potential plagiarism in the codes between the students. The copy checker is made to be very reliable so that it is highly likely to mark a pair of code as a copy even though two students quickly discuss the idea without looking at each other's code. Of course, we will evaluate the similarity of a pair compared to the overall similarity for the entire class.
- We will also inspect the codes manually. In case we doubt that the code may be written by someone else (outside of the class), we reserve the right to request an explanation about the code. We will ask detailed questions that can only be answered when the codes were written by yourself.
- If one of the above cases happen, you will get 0 marks on the assignment and may get a further penalty. Please understand that we apply these methods for the fairness of the assessment.
- Download and unzip "HW1.zip" file from the ETL. "HW1.zip" file contains skeleton codes for Question 1 ~ 10 (in the "problem1" ~ "problem10" directory respectively)
- Do not modify the overall directory structure and Test.java after unzipping the file and fill in the codes in appropriate files. It is okay to add new directories or files if needed.
- Once you have completed your homework, compress the "HW1" directory in a single zip file named "**20XX-XXXXX.zip**" (your student ID) and upload it to ETL as you submit the solution for the lab tests. Contact the TA if you are not sure of the process. Make sure to double-check if your final zip file is properly submitted.
- Do not use any external libraries. JAVA Collection Framework (List, Set, Map, Queue, ...) is also not allowed.
- You don't have to consider Wrong Inputs, and the trailing spaces will be ignored.
- You can change all other classes except for Test.java (test will be done by the main method in Test.java)

Submission Guidelines

Example Submission Code Directory Structure (after unzip)



Submission Directory Structure

- HW1
 - problem1
 - Test.java (**DO NOT modify**)
 - ... (You can create other directories or files.)
 - problem2
 - Test.java (**DO NOT modify**)
 - ... (You can create other directories or files.)
 - ... (Same for other problems)
1. **You should submit your code on eTL.**
 2. **After you extract the zip file, you must have a HW1/ directory.**
 3. You can create other directories or files in each problem directory.
 4. Compress the “HW1” directory to “20XX-XXXXX.zip” (your student ID)
 5. This is different from the IntelliJ directory structure, so move the source file in the src/ directory to problemX/ directory.

Assignment 1.

1. Implement simple Java programs.
2. All inputs/outputs are console input/output.
3. Problem 9, 10 are more difficult than problem 1 ~ 8.
4. You can use Autolab. (See ‘Simple Autolab Guide.pdf’ on eTL)

Question 1: Multiplication Table [0.5 Marks]

Objective: Write a program that prints out the multiplication table corresponding to the input number N. ($1 \leq N \leq 30$)

Note:

1. Outputs always start with '1 times 1 = 1'
2. Outputs will be $N \times N$ lines

Target Function: `public static void printMultiplicationTable(int n)` (in MultiplicationTable.java)

Input: Integer N ($1 \leq N \leq 30$)

Output: Multiplication table corresponds to input number N.

Input	Output
3	1 times 1 = 1 1 times 2 = 2 1 times 3 = 3 2 times 1 = 2 2 times 2 = 4 2 times 3 = 6 3 times 1 = 3 3 times 2 = 6 3 times 3 = 9

Question 2: Fibonacci Numbers [0.5 Marks]

Objective: Write a program that prints out N smallest Fibonacci numbers. ($1 \leq N \leq 30$)

Description: Fibonacci Numbers, commonly denoted as F_n , form a sequence called the Fibonacci sequence. The sequence starts with the two numbers, 0 and 1, and each number in the sequence is the sum of its two preceding ones, that is, $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ for $n > 1$. Print $F_0 \sim F_{N-1}$.

Reference: https://en.wikipedia.org/wiki/Fibonacci_number

Target Function: `public static void printFibonacciNumbers(int n)` (in FibonacciNumbers.java)

Input: Integer N ($1 \leq N \leq 30$)

Output: The first N Fibonacci Numbers from the smallest to the largest.

Input	Output
5	0 1 1 2 3

Question 3: Drawing Figure [0.5 Marks]

Objective: Write a program for the input number N to print out the top N-line of the figure given below. ($1 \leq N \leq 6$)

```
////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
////////////////////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
////////////////////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
//////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
//////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
//////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
*****
```

Target Function: `public static void drawFigure(int n)` (in DrawingFigure.java)

Input: Integer N ($1 \leq N \leq 6$)

Output: Top N-lines of the figure.

Input	Output
2	<pre>////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ ////////////////////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\</pre>
6	<pre>////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ ////////////////////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ ////////////////////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ //////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ //////////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ //////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ //////*****\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ *****</pre>

Question 4: Character Counter [0.5 Marks]

Objective: Write a program that

1. takes in an arbitrary string composed only of lowercase alphabets ($1 \leq \text{input string length} \leq 200$)
2. counts the number of alphabets that are in that string
3. prints out the alphabet and the corresponding count in an increasing character order. (DO NOT print out alphabets that have never appeared in the string).

Target Function: `public static void countCharacter(String str)` (in CharacterCounter.java)

Input: String str (1 <= length <= 200)

Output: Character counts with increasing character order.

Input	Output
bbbaabacechadhbcbhhdcbbehah	a: 5 times b: 8 times c: 4 times d: 2 times e: 2 times h: 6 times

Question 5: Number Counter [0.5 Marks]

Objective: Write a program which counts the frequency of each number (0~9) in the multiplication of three input numbers.

Description:

1. There are three input numbers. Each number is in range $1 \leq N \leq 999$.
(ex. 214, 731, 200)
2. Multiply the input numbers.
(ex. $214 \times 731 \times 200 = 31,286,800$)
3. Count the frequency of each number (0~9) from the multiplied result.
(0: 2 times, 1: 1 Times, 2: 1 Times, 3: 1 Times, 6: 1 Times, 8: 2 Times)
4. Print out the counted frequency in the increasing order of the numbers.
(DO NOT print out numbers that have never been counted)

Target Function: `public static void countNumbers(String str0, String str1, String str2)` (in NumberCounter.java)

Input: Three numbers per each input line.

Output: Number counts with the increasing order of the numbers. DO NOT print out numbers that have never been counted.

Input	Output
927 281 350	0: 2 times 1: 2 times 4: 1 times 5: 1 times 7: 1 times 9: 1 times

Question 6: Prime Numbers [0.5 Marks]

Objective: Write a program that outputs prime numbers as many as the input number N.
($1 \leq N \leq 100$)

Description: A prime number (or a prime) is a natural number greater than 1 that is not a product of two smaller natural numbers. The smallest prime number is 2, and the second minority is 3. Print N prime numbers in ascending order.

Reference: https://en.wikipedia.org/wiki/Prime_number

Target Function: `public static void printPrimeNumber(int n)` (in PrimeNumbers.java)

Input: Natural number N ($1 \leq N \leq 100$)

Output: N Prime Numbers from the smallest to the largest.

Input	Output
5	2 3 5 7 11

Question 7: Increasing String [0.5 Marks]

Objective: Write a program that finds the longest substring where each alphabet in the substring is in ascending order.

Description: A substring is a contiguous sequence of characters within a string. Among substrings, there are increasing substrings, where the alphabet on the right is always bigger than the one on the left. Find the length of the longest increasing substring. If a substring contains multiple identical alphabets, the substring is not an increasing substring.

(ex. Input: "abcddeacefzbce" \Rightarrow Longest increasing substring: "acefz" \Rightarrow Output: 5)

(Note. "abcdde" is NOT an increasing substring.)

Target Function: `public static int printLongestIncreasingSubstringLength(String inputString)`
(in IncreasingString.java)

Input: A string which consists of small letters. ($1 \leq \text{input string length} \leq 200$)

Output: Length of the longest increasing substring.

Input	Output
abcdacea	4
abcddeacefzbce	5
abab	2

Question 8: Matrix Flip [0.5 Marks]

Objective: Write a program that flips (upside down and left to right) the input matrix. For example, given a 3 x 3 input matrix on the left, generate the output matrix on the right.

Input Matrix	Output Matrix
AAA ABB BBB	BBB BBA AAA

Description: Implement a method that prints out the flipped matrix.

Target Function: `public static void printFlippedMatrix(char[][] matrix)` (in MatrixFlip.java)

Input: The first line of the input contains two integers n and m where n is the number of rows and m is the number of columns. n and m are separated by a whitespace. Assume that $1 \leq n \leq 100$, and $1 \leq m \leq 100$. The matrix values are from the second line to $(n + 1)^{\text{th}}$ line.

Output: Flipped matrix

Input	Output
3 3 AAA ABB BBB	BBB BBA AAA
4 5 BBBBBC CBBBBB BBBBBB CCCCC	CCCCC BBBBBB BBBBBC CBBBBB

Question 9: Fractional Number [0.5 Marks]

Objective: Write a program that calculates the input equation, which consists of four fundamental arithmetic operations.

Description:

1. The input equation will be composed of two numbers and an operator.
2. Two numbers and an operator are separated by spaces.
3. The input number can be two types.
 - a. Fractional number: two natural numbers with '/' between them (ex. 2/3)
 - b. Natural numbers (ex. 1, 2, 3, ...) ($1 \leq N \leq 999$)
4. There are 4 operators: '+', '-', '*', '/'
5. All input numbers are positive, but the result can be negative.
6. The result should be an irreducible fraction. (You can use `FractionalNumber.gcd()`)
7. If the result is negative, put '-' first. DO NOT make the denominator minus.
(-2/3 \Rightarrow Good, 2/-3 \Rightarrow Bad)
8. If the result is a natural number, DO NOT use '1' as the numerator.
(3 \Rightarrow Good, 3/1 \Rightarrow Bad)

Useful Resources:

- String: <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html>

Target Function: `public static void printCalculationResult(String equation)` (in `FractionalNumber.java`)

Input: Equation to solve. Format: <Number><Space><Operator><Space><Number>

Output: Calculated result. The result should be an Irreducible Fraction.

Input	Output
2/3 + 3/4	17/12
6/9 - 2	-4/3
1/2 + 1/2	1

Question 10: Card Game Simulator [0.5 Marks]

Objective: Write a program that simulates a card game.

Description:

1. There are 20 unique cards with 10 numbers (0 ~ 9) and 2 shapes (O, X).
2. Player A and B starts the game with 10 cards each. (Input contains this information)
 - a. Input consists of 2 lines.
 - b. The first line indicates Player A's card list, and the second line indicates Player B's card list. (See Example Input and Output)
3. The two players take turns playing cards.
4. The players choose the card to play with the following rules.
 - a. The game starts with Player A. Player A uses the card with the largest number. If there are two cards (O, X) with the largest number, Player A will choose the card with shape O among the two cards.
 - b. After that, if the next player has a card that has the same number as the previous player's, that card is used.
 - c. If the next player does not have a card that has the same number as the previous player's but has the same shape card, the player will choose the same shape card with the largest number.
 - d. If the next player does not have both of the same number cards and the same shape cards, the current player wins the game.
 - e. If the two players used all of the cards, Player B wins.
5. Simulate the game by printing the sequence of the cards.
6. Print out the result message when the winner is decided.

Useful Resources:

- String: <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html>
- PrintStream: <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/PrintStream.html>
(Type of System.out is PrintStream)

Target Function: `private static void simulateCardGame(String inputA, String inputB)` (in CardGameSimulator.java)

Input: The first line \Rightarrow Player A's card list, The second line \Rightarrow Player B's card list

Output: Simulated card game progress and result.

Input	Output
00 3X 40 4X 7X 10 6X 50 60 9X 2X 20 30 90 5X 1X 80 70 8X 0X	Player A: 9X Player B: 90 Player A: 60 Player B: 80 Player A: 50 Player B: 5X

	Player A: 7X Player B: 70 Player A: 40 Player B: 30 Player A: 3X Player B: 8X Player A: 6X Player B: 2X Player A: 4X Player B: 1X Player A: 10 Player B: 20 Player A: 00 Player B: 0X Player A loses the game!
9X 30 1X 10 80 3X 00 0X 50 6X 60 20 5X 90 8X 2X 4X 40 70 7X	Player A: 9X Player B: 90 Player A: 80 Player B: 8X Player A: 6X Player B: 60 Player A: 50 Player B: 5X Player A: 3X Player B: 7X Player A: 1X Player B: 4X Player A: 0X Player B: 2X Player A loses the game!
00 10 20 30 40 50 60 70 80 90 0X 1X 2X 3X 4X 5X 6X 7X 8X 9X	Player A: 90 Player B: 9X Player A loses the game!