(Lab 2) Problem Set 2:
Control Structures

P2 Solutions limited in scope to:		
● P1 Concepts	● Selection statements ○ single selection ○ double selection ○ multi-selection	● Repetition statements ○ counter controlled loops ○ sentinel controlled loops

Submission Rules:

1. Submissions must be zipped into a **handin.zip** file. Each problem must be implemented in its own class file. Use the name of the problem as the class name.

2. You must use standard input and standard output for ALL your problems. It means that the input should be entered from the keyboard while the output will be displayed on the screen.

3. Your source code files should include a comment at the beginning including your name and that problem number/name.

4. The output of your solutions must be formatted exactly as the sample output to receive full credit for that submission.

5. Compile & test your solutions before submitting.

6. Each problem is worth up to 10 points total. The breakdown is as follows: 2 points for compiling, 3 points for correct output with sample inputs, 5 points for additional inputs.

7. This lab is worth a max total of: 40 points. You can complete as many problems as you like, but cannot receive more than 40 points towards the lab grade. All points in excess of that are for bragging rights. (Check the scoreboard to see how you did!)

8. Submission:

- You have unlimited submission attempts until the deadline passes

- You'll receive your lab grade immediately after submitting

- **IMPORTANT:** if your grade is lower than 70% when the deadline passes, then you must attend a recitation session & get TA signoff to receive full credit for that lab challenge.

- **Online help/discussions**: www.acmuno.slack.com, **channel: *#1583-java1***

Problem 1: Summing It Up

Write a program, which takes two distinct integers separated by space as input and prints the sum of all the integers between them, including the two given numbers. Note that the numbers can appear in either order. You may assume that both numbers are between
–10, 000 and 10, 000.

For example, if the input is as follows:
10 4
the output should be 49, since 10+9+8+7+6+5+4=49.

Similarly, if the input is
-3 10

the output should be 49,
since (-3) + (-2) + (-1) + 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 49

Input

The input will begin with a single line containing T , the number of test cases to follow. The remaining lines contain the lines to be calculated. Each of these lines has two integers separated by a single space.

Output

The output should consist of the sum of all numbers between the two input numbers.

Sample Input	Sample output

2
10 4
-3 10
	49
	49

Problem 2: Point of Sale System

The McDowell Restaurant chain has asked you to write a menu program for their new Fast-food service machines. Your program already prints the following menu like this:

```
**********************************************************************
                            McDowell’s Restaurant

**********************************************************************

Make your selection from the menu below:

1. Regular Hamburger $1.50
2. Regular Cheeseburger $1.75
3. Fish Sandwich $2.50
4. Half-pounder with cheese $2.75
5. French Fries $0.99
6. Large Soft Drink $1.25

***********************************************************************
Select 1, 2, 3, 4, 5, or 6 ----- >
```

Your program must now read the customer’s selection and compute the total price of their purchase, including 6.5% sales tax.

Input
The first line of input represents N, the number of test cases. The additional lines consists of a sequence of integers scoped between 1 to 6. Each number should indicate a selection from the above menu to be purchased.

Output
The program should print the sum of the orders plus tax as “Please pay $<dollars>”, where <dollar> is the total amount of the purchase then conclude with “Thank you for eating at McDowell’s”.

Sample Input	Sample Output
1 1 4 4 5 3 1	Please pay $12.77 Thank you for eating at McDowell’s!

Problem 3: An Interesting Problem

Write a program that accepts two positive integers: a deposited amount of money and an interest rate, as an annual percentage rate. Your program will calculate the number of years that will take for the account balance to reach $1, 000,000. You can assume that the initial deposit is less than $1,000,000

Input

The input will begin with a single line containing T , the number of test cases to follow. The remaining lines contain the lines to be calculated. Each of these lines has two positive integers separated by a single space. The first value is the deposited amount, the second is the interest rate.

Output

The output should consist of the number of years.

Sample Input	Sample output

2
10000 10
500 5
	49 years
	156 years

Problem 4: Min/Max Search by Value

Develop a program that, given a sequence S of integers as input, produces as two output values, the first is the minimum value that appears in the sequence and the second is the maximum value that appears in the sequence.

Facts

- Scanner has a method that returns a boolean indicating whether a next integer exists in its input stream (`hasNextInt()`)
- Scanner objects can be initialized to to scan String data as input.

Input

The input will begin with a single line containing T , the number of test cases to follow. The remaining lines contain the T sequences, one line per sequence. Each of these lines contains the values in the sequence. Each such value is separated from the next by at least one space.

Output

For each sequence given as input, there should be four lines of output. The first line echos the given sequence. The second line indicates the minimum value that occurs. The third line indicates the maximum value that occurs. The fourth line is blank.

Sample Input	Sample Output

3
3 6 -1 4 6 5 3
0 0 0 0
-4 45 2 0 3 5 11 -7 854 25 3 -7 4 -3
	3 6 -1 4 6 5 3
	-1
	6
	
	0 0 0 0
	0
	0
	
	-4 45 2 0 3 5 11 -7 854 25 3 -7 4 -3
	-7
	854

Problem 5: Min/Max Search by Index

Develop a program that, given a sequence S of integers as input, produces as output two sequences of positive integers, the first of which indicates all those positions in S at which S's minimum value occurs and the second of which indicates all those positions at which S's maximum value occurs. Positions are numbered starting at zero (0).

Facts

- Scanner has a method that returns a boolean indicating whether a next integer exists in its input stream (`hasNextInt()`)
- Scanner objects can be initialized to to scan String data as input.

Input

The input will begin with a single line containing T , the number of test cases to follow. The remaining lines contain the T sequences, one line per sequence. Each of these lines contains the values in the sequence. Each such value is separated from the next by at least one space.

Output

For each sequence given as input, there should be four lines of output. The first line echos the given sequence. The second line indicates the positions at which the minimum value occurs. The third line indicates the positions at which the maximum value occurs. The fourth line is blank.

Sample Input	**Sample Output**

3
3 6 -1 4 6 5 3
0 0 0 0
-4 45 2 0 3 5 11 -7 854 25 3 -7 4 -3
	3 6 -1 4 6 5 3
	2
	1 4
	
	0 0 0 0
	0 1 2 3
	0 1 2 3
	
	-4 45 2 0 3 5 11 -7 854 25 3 -7 4 -3
	7 11
	8

Problem 6: Chair Scavenging

Your computer lab often runs out of chairs to sit in during programming practices. In order to overcome this hardship, the team borrows chairs from other adjacent rooms. Given the number of people who are coming to practice and the number of chairs in the room, how many more chairs must the team take from other rooms? Sometimes there are enough chairs to begin with. In this case, we need not take any more chairs to the room.

Input

The input will begin with a single line containing T , the number of test cases to follow. On each of the next T lines there will be two integers, N (1 ≤ N ≤ 10) and M (0 ≤ M), separated by a single space. N is the number of people coming to practice that day and M is the number of chairs already in the room that day.

Output

The output will consist of T lines, each containing a single number the number of additional chairs needed for that day.

Sample Input	**Sample Output**

2
10 9
15 8
	1
	7

Problem 7: Rock Paper Scissors (10 points)

(Game Development) Rock, Paper, Scissors is a hand-based game played between two people. During a match, each player chooses one of three possible options: Rock, Paper, or Scissors. Afterwards, both players reveal their selection simultaneously. Your task is to implement a program that acts as a referee *(i.e. game manager)* to determine the winner of a rock-paper-scissors game.

Facts

- Rock beats Scissors
- Scissors beats Paper
- Paper beats Rock
- If both player selections match then its a tie.

Input

The first line is the number of test cases. Then each additional line will contain two different text inputs. The first text input represents the first player's selection. The second text input represents the second player's selection. Each player selection is limited to: "paper", "rock", "scissors"

Output

Your program must display the final result of that game as on of the following: "Tie!", "Player 1 wins!" or "Player 2 wins"

Sample Input	Sample Output
3 paper paper rock paper scissors paper	Tie! Player 2 wins! Player 1 wins!

Problem 8: DNA Pairing (10 points)

(Bioinformatics) DNA (DeoxyriboNucleic Acid) is present in nearly all living organisms as the main constituent of chromosomes. It is the carrier of genetic information. DNA typically consists of two separate strands paired together in the form of a double helix. Each strand contains the nucleobases: “A” (adenine), “G” (guanine), “C” (cytosine) and “T” (thymine). Hydrogen bonds hold the two strands of DNA together, where base A always pairs with T and G always pairs with C. Write a program that takes in a segment strand of a DNA sequence and outputs it’s complementary strand.

Facts

- Scanner has a method `useDelimiter` that alters the break point for pulling text from input stream. To grab 1 letter at a time, use empty string as delimiter *i.e.* `useDelimiter("")`
- Scanner has a method that returns a boolean indicating whether a next integer exists in its inputstream (`hasNext()`)
- Scanner objects can be initialized to to scan String data as input.

Input

The first line is the number of test cases. Then each additional line contains a DNA sequence of nucleobases that comprise a segment of one strand of DNA. Bases include "A", "C", "G", "T"

Output

Your program must display the original sequence of a DNA stand along with it’s appropriately matched compliment pair.

Sample Input	**Sample Output**
3 AGAATTGCATAG GGATTCACATAG ATGCATGCATGC	AGAATTGCATAG TCTTAACGTATC GGATTCACATAG CCTAAGTGTATC ATGCATGCATGC TACGTACGTACG