Contest Problems
Philadelphia Classic, Spring 2018
Hosted by the Dining Philosophers
University of Pennsylvania

dining philosophers

UNIVERSITY OF PENNSYLVANIA | COMPUTER SCIENCE CLUB

Rules and Information

This document includes 12 problems. Novice teams do problems 1-8; standard teams do problems 5-12.

Any team which submits a correct solution for any of problems 1-4 will be assumed to be a novice team. **If you are not a novice team, please skip problems 1-4.**

Problems 1-4 are easier than problems 5-8, which are easier than problems 9-12. These problems are correspondingly labeled “Novice”, “Intermediate”, and “Advanced.” Order does not otherwise relate to difficulty.

You may use the Internet only for submitting your solutions, reading Javadocs or Python documentation, and referring to any documents we link you to. You **may not** use the Internet for things like StackOverflow, Google, or outside communication.

As you may know, you can choose to solve any given problem in either **Java or Python**. We have provided stub files for all questions in both languages that takes care of the input parsing for you. **Do not modify any of the parsing or output code**. Just fill out the stub methods in each file and submit it with exactly the same name.
Do not modify any of the given methods or method headers in our stub files! They are all specified in the desired format. You may add class fields, helper methods, etc as you like, but modifying given parts will cause your code to fail in our testers.

There is no penalty for incorrect submissions. You will receive 1 point per problem solved. A team’s number of incorrect submissions will be used only as a tiebreaker.

Some problems use Java’s “long” type; if you are unfamiliar with them, they’re like an “int”, but with a (much) bigger upper bound, and you have to add “L” to the end of an explicit value assignment:

```
    long myLong = 1000000000000L;
```

Otherwise, the “long” type functions just like the “int” type.

1. A world in reverse

A group of people from Phildystopia have been held captive by a malicious trickster called Dysonius. He wants to take over their identities, send them into an alternative reality and wreck havoc in everyone’s life. It’s up to you to save them and set them free! To be “fair”, the trickster has given you a clue but you must decipher what it means as the trickster does not speak English. To do this, you must reverse the letters of the words in the clue and look up what the reversed sentence means in the dictionary that is kept in the City Council House. You must tell the trickster you have cracked his clue otherwise he won’t set them free. Hurry before the people disappear forever!

Input Format

A string **S** consisting of one or more than one word. The words may contain letters of the alphabet in upper or lower case or any other special characters.

Output Format

A string with the letters of the words of **S** reversed.

Sample Input	Sample Output	Explanation
What goes around, comes around!	tahW seog ,dnuora semoc !dnuora	Each word separated by a space is reversed. This also includes special characters.
Hi	iH	A word with no spaces reversed. The first letter is now the last letter and the last letter, the first. Uppercase and lowercase characters are maintained.

2. Evil Merger

The people of the city of Phildystopia have been enjoying entertainment from various different companies. However, two major firms, NT&T and Dystime Warner, have decided to merge in order to increase their reach and supply entertainment to a larger number of people. However, it is speculated that this merger may result in them extracting much higher prices from the people!

In the absence of any laws, as they plan this merger, their evil competitor Dysflix must evaluate how this can be beneficial to itself. It has collected the following data from the two evil companies about their pricing strategy once they merge.

Your job, as head researcher for Dysflix, is to find out the correlation between the prices and number of customers using the site of the merged firm so you can evaluate whether that would mean more customers for you!

You can use this formula to calculate the correlation:

$$\frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{s_x}\right)\left(\frac{y_i-\bar{y}}{s_y}\right)$$

Formula to find the standard deviation s_x and s_y:

$$s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

$$s_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i-\bar{y})^2}$$

Input Format

Two lists of integers, **A** and **B**.

Output Format

A real number between -1.00 and 1.00 (specified to 2 decimal places), denoting the correlation between the two lists **A** and **B**.

Sample Input	Sample Output	Explanation
[1, 2, 3, 4] [10, 8, 6, 4]	-1.00	Here n = 4. The average of the first list is 2.5 and the average for the second list is 7.0. The standard deviation of the first list is 1.29 and the standard deviation for the second list is 2.58. Using the formula above, we have correlation = -1.00.
[10, 8, 6, 4] [4, 3, 2, 1]	1.00	Here n = 4. The average of the first list is 7.0 and the average of the second list is 2.5. The standard deviation of the first list is 2.58 and the standard deviation of the second list is 1.29. Using the formula given we have, correlation = 1.00.

3. PhilCoin Frenzy

Having made it big in Philutopia, you decide to share your riches with your family back home. However, all your money is in the form of Philutopia's revolutionary new digital currency PhilCoin, which is not yet recognized by the rest of the world. After much searching, you finally find a coin exchange willing to convert your PhilCoin to USD. Unfortunately, the exchanger is only willing to convert particular sums of PhilCoin. Specifically, he requires that after squaring the number of PhilCoins, the resulting number can be split into two parts that sum to your original balance. The first part is returned to you in PhilCoin, and the second part will be converted to USD.

You open your virtual wallet and check your current balance. Can you figure out if you can exchange your PhilCoin such that you receive more than 0 USD?

Input Format
A non-negative integer, **n**.

Output Format
True if the square of the input can be split into two parts that sum to the input, and the second part is nonzero
False otherwise

Sample Input	**Sample Output**	**Explanation**
1	True	1^2 = 01: 0 + 1 = 1.
14	False	14^2 = 196: neither 1 + 96 nor 19 + 6 is equal to 14.
10	False	10^2 = 100: though 10 + 0 indeed equals 10, this ignores the constraint that the second part may not be 0.

4. Going the Distance

The city of Phildystopia has recently launched a new venture that allows them to manufacture robotic cyborgs, which they plan to use to invade Philutopia. While planning their invasion, Phildystopian scientists do realize that they need to test their cyborgs in a controlled environment before they are unveiled to the world.

Each cyborg is born with a set of its own DNA. The technology that they use allows them to specify a robotic cyborg's DNA as a binary sequence in where each element in the sequence indicates the presence or absence of a gene. A 1 indicates that the corresponding gene is present in the robotic cyborg, whereas a 0 indicates that the corresponding gene is absent in the robotic cyborg. For every cyborg, scientists tag each one with a DNA identifier, which has the value of the DNA written in base 10. For example, if cyborg A had the DNA identifier 11, then its DNA sequence would be 1 0 1 1, since 13 written in binary is 1011_2.

The scientists would like to test the cyborgs by comparing their DNA in pairs. Specifically, they are interested in cyborgs with significantly different DNA sequences. The difference in the DNA sequences of two cyborgs is measured by calculating the total number of genes that are different in both sequences. Can you help the scientists implement the tester and invade Philutopia?

Input Format
Two integers **A** and **B** that represent the DNA identifiers of cyborgs A and B, respectively.

Output Format
A single integer representing the number of places their DNA sequences differ.

Sample Input	**Sample Output**	**Explanation**
1 4	2	1 → 0 0 0 1 4 → 0 1 0 0 There are precisely 2 locations that differ.
1 1	0	1 → 0 0 0 1 1 → 0 0 0 1 The DNA are identical.

5. Gold Rush

The chief investment bank of Philutopia, Soldman Gachs, has a massive vault of gold bars hidden underground. There are **N** doors, with door i containing N_i gold bars labeled with the number i (1 <= i <= **N**). Each door contains less than 10 gold bars. Unfortunately, a Storgan Manley investment banker from Phildystopia found the passcode for the vault, and plans to mess with Soldman Gachs' operations! He has limited time, and can't carry any gold out of the vault, so he thinks the best prank he can play is to swap two sets of gold bars behind two different doors, and he wants to maximize the amount that Soldman Gachs thinks they have inside the vault with one swap. Note that if the evaluation is already maximized, he will feel generous and decide not to make any changes to the vault.

Lerill Mynch, senior analyst of Soldman Gachs, comes up with a daily evaluation as follows: every gold bar behind door i is worth $10^{\mathbf{N} - i}$ dollars to Lerill, and the sum of the values of gold bars is his final evaluation. After the vault heist was completed, what is the maximum evaluation that Lerill could come up with?

Input Format

A list of **N** integers between 0 and 9 (inclusive), where the ith integer represents the number of gold bars behind door i. **N** can range from 1 to 1000.

Output Format

The maximum possible evaluation that would result after the gold bars are swapped (if any are swapped). As this number may be big, **output the number as a string**.

Sample Input	**Sample Output**	**Explanation**
[1, 2]	21	The gold bars of the 2 doors can be swapped to yield a max evaluation of 21.
[8, 5, 1]	851	Note that the evaluation can't be increased via any swap, so the max evaluation is just 851.
[1, 2, 3]	321	The gold bars of doors 1 and 3 can be swapped to yield a max evaluation of 321.

6. Quick Maths

The city of Philutopia could not be better in terms of mathematics education. Students are aceing all the exams, and are quicker at calculations than anyone could have imagined. However, the teachers are having trouble keeping up with the students! To this end, the city has commissioned you to help these teachers keep up with their students.

The city hands you a list of hundreds of mathematical expressions for you to quickly compute. Your job is to write a function that takes in a single mathematical expression and outputs its evaluation. The expression may contain numbers (0-9), addition signs (+), subtraction signs (-), multiplication signs (*), division signs (/), or parentheses (()). You may assume that the expression is well-formed, i.e. all parentheses are matched and the expression is valid. Furthermore, assume that all outputs are integers, divisions will never yield a non-integral value, and there are no divisions by zero.

Can you help implement this program for the schools of Philutopia?

Input Format

A string consisting of a mathematical expression, with operators separated by a single space. The length of the expression **N** is in the range 1 <= **N** <= 100000. The expression contains no spaces.

Output Format

Output a single integer representing the evaluation of the expression.

Note: Any solution involving the use of a built-in evaluation function such as Python's eval() function will receive ZERO points.

Sample Input	**Sample Output**	**Explanation**
(2 + 2) - 1	3	2 + 2 = 4, and 4 - 1 = 3.
2 + 2 * 3	8	Note that order of operations is followed in the expression.
((3 + 4) * (6 + 3)) - 2	61	The expression evaluates to 61 when manually calculated.

7. Maximum Sum

Staying in Phildystopia is no longer an option—the life is just not up to your standards. Thus, you decide to plan an escape. However, to fund your escape, you need to steal money. A spy working for the government hands you a list containing amounts of money stored at certain locations, in order of proximity to you. However, certain locations may have negative values, since in order to enter the location you'll need to pay a fee. You want to maximize the amount of money you steal, while at the same time not stealing money from any two locations that are next to each other on the list (to make it harder for the police to catch you). Can you write a program that finds the maximum amount of money you should steal? (Note that you have the option to steal no money.)

Input Format
A list of integers **A** representing the amount of money you can get from each location.

Output Format
Output a single integer representing the maximum amount of money you can get without stealing from adjacent locations in the list.

Sample Input	**Sample Output**	**Explanation**
[1, 2, 3, 5]	7	The maximum sum of non-consecutive elements is 2 + 5 = 7.
[1, 100, 10]	100	The maximum sum of non-consecutive elements is 100.
[-5, -3, -2, -1]	0	The maximum sum of non-consecutive elements is -3 + -1 = -4. However, since -4 < 0, return 0.

8. Robotic Riddles

During your travels through Phildystopia you often run into groups of abandoned robots that can help point you towards your destination. However, these robots are often a little bit buggy and can sometimes point you in the wrong direction. Whenever you encounter a group of robots, each robot will give you **two** instructions. The two types of instructions a robot can give you are either a **positive number**, representing a "path" that you should travel down next, or a **negative number** representing a "path" that you absolutely should not travel down next. Your goal is to determine, based on all the instructions given by each robot in a group, if any of the robots gave you two instructions that conflict. In other words, you want to know if every robot gave at least one instruction that could be accurate. Keep in mind there are many paths that you could take next to reach your destination, and a robot may give you the same instruction twice rather than two different instructions.

Input Format

An array **A** of arrays of length 2, such that the ith array contains the 2 instructions from Robot i.

Output Format

False if the robot group has a robot that gave two bad instructions, and True otherwise

Sample Input	**Sample Output**	**Explanation**
[[1, 1], [2, 2], [-1, -2]]	False	Robot 1 tells you to take path 1, Robot 2 tells you to take path 2, and Robot 3 tells you not to take paths 1 or 2. If Robot 3's first instruction is good, then both of Robot 1's instructions are bad. If Robot 3's second instruction is good, then both of Robot 2's instructions are bad. Thus, at least one robot gave two bad instructions.
[[1, 2], [-2, 3], [3, 1]]	True	If paths 1 and 3 are good, then each robot provided at least 1 instruction that is accurate without any conflicts arising.

9. Parallelogram Lock

With the robot uprising, humanity lost its position as the controllers of the world. Fortunately, you are one of the survivors and you have joined the resistance to fight for a place for the living. Unfortunately though, the security of the robots is very tight and most of the attacks made by the resistance resulted in failure. However, one final hope arrived, the resistance intercepted one message detailing the next security protocol the robots are going to put on their facilities: a parallelogram lock. The idea is simple: the lock details a number of points in the plane and the password is the number of parallelograms that can be formed with these points. Robots can solve this problem very easily with their perfect counting skill, however humans frequently lose the count, truly a mischievous scheme. But now you can prepare: write a program that counts the number of parallelograms for you and help the resistance triumph.

Input Format

The first line contains an integer n representing how many points are there in the plane.
Each following n lines contains exactly numbers, the x and y coordinates of one of the points.

Output Format

A single integer representing the number of possible parallelograms that can be formed from the given points on the plane.

Sample Input	Sample Output	Explanation

3
1 1
-1 3
7 -1
	0	Parallelograms have 4 points, so of course we can't have any if we only have 3 points.

4
1 1
1 0
0 1
0 0
	1	We have exactly one parallelogram, in fact it is even a square!

4
0 1
0 2
0 3
0 4
	1	We count degenerate parallelograms, that is, those were the four points are collinear, as parallelograms. To better understand this case, remember that

		opposite sides of a parallelogram have the same size and are parallel.

24
0 0
0 1
0 2
0 3
0 4
1 0
1 1
1 2
1 3
1 4
2 0
2 1
2 3
2 4
3 0
3 1
3 2
3 3
3 4
4 0
4 1
4 2
4 3
4 4
	574	The patterns can become very tricky and hard for humans to count parallelograms, good luck! Hint: this is not the whole grid of points with 0<=x<=4 and 0<=y<=4.

10. Hive Minds

The invasion was successful! The resistance was able to infiltrate one of the robot hive minds controllers and now it can finally implement its plan to overthrow the robot hegemony. To do so, they need a very important information: at any mind they might need to know either two robots are in the same hive mind in order to know if it is safe to attack them. Luckily, you got access to the live log of the hive mind controller and can help them answer these questions!

Input Format

The first line contains an integer n representing how many operations you are going to process. Each following n lines contains exactly 3 numbers, the type of the operation and the numbers of two robots.

An operation of type 0 represents a merging between two hive minds. Namely, the next two elements of the tuple will contain numbers R_1 and R_2 representing that the hive minds in which robots R_1 and R_2 are will be merged. R_1 and R_2 can be the same, in this case do nothing.

An operation of type 1 represents a robot leaving a hive mind and joining another. Namely, the next two elements of the tuple will contain numbers R_1 and R_2 representing that the robot R_1 will leave his hive mind and join the hive mind of robot R_2. R_1 and R_2 ca be the same, in this case the robot leaves his previous hive mind and form an unitary hivemind containing only him.

An operation of type 2 represents question from the resistance. Namely, the next two elements of the tuple will contain numbers R_1 and R_2 representing that the resistance wants to know either robots R_1 and R_2 are in the same hive mind. R_1 and R_2 can be the same, in this case the answer is always “True”.

Output Format

Return a list of booleans containing the answers to the questions from the resistance (operation type 2), we say that the answer is “True” if the robots are in the same hive mind and “False” if they are not. Note that the length of this list has to coincide with the number of operations of type 2 in the input.

Sample Input	**Sample Output**	**Explanation**

3
2 0 1
0 0 1
	False
	True
		Initially the robots are all i their own unitary hive minds so the first answer is “FALSE”, however we joined robot 0 and robot 1 in the

2 0 1		second operation, so the answer to the question in the second operation is “TRUE”.

7
0 0 1
0 2 3
2 0 3
1 0 2
2 0 3
1 3 1
2 0 3
	False
	True
	False
		We join 0 and 1 then 2 and 3, so our hive minds are {0,1} and {2,3} when we ask the first question, so the answer is “FALSE”. Then, we start moving some robots around. The fourth operation brings us to {1} and {0,2,3}. Thus, the answer to the second question is “TRUE”. Then, we perform yet another move, bringing us to {1,3} and {0,2}. Hence the answer to the last question is “FALSE”.

11. Self-Driving Car Navigation

Philutopia is testing out its self-driving car program. They have set up an elaborate track with obstacles, few of which have to be avoided by the car in order to complete the track. Each obstacle has a car-detection range, and any car in that range fails the track. However, the deep learning models that power the car are extremely sophisticated, and they can avoid the obstacles one by one. However, these models need a little more training, and they must be trained efficiently.
The track is represented by a rectangle on the x-y plane with bottom-left corner (0, 0) and top-right corner (width, height). The self-driving cars start at the left end of the track (x = 0) and are trying to move to the right end (x = width). The track is bounded by lines from (0,0) to (width, 0) and from (0, height) to (width, height). Can you help train the models by calculating the minimum number of obstacles which need to be avoided in order for the car to complete the track?

Input Format
The first line contains the width and height. The second line contains the number of robots, **N.**
The next **N** lines contain information pertaining to each obstacle: the x coordinate, the y-coordinate, and the car-detection range.

Output Format
You should output a single integer denoting the minimum number of obstacles that need to be avoided by the car.

Sample Input	**Sample Output**	**Explanation**

20 4
1
10 2 2
	1	There is one obstacle at (10, 2) with detection range 2, which the car needs to avoid to get through the track.

20 4
1
10 1 2
	0	There is a gap through which the car can reach the end, so no obstacle needs to be avoided.

20 4
3
10 1 2
	1	Only the third obstacle needs to be removed in order to get through the

11 1 2 10 3 4		track.

12. The Final Escape

The end is in sight. You are on your final quest to escape Phildystopia. In front of you is a number **M,** and a cipher with **N** locks, labeled from 1 to **N.** Each lock has a secret number that is used to unlock it. The secret number is generated using the following rule: for lock labeled **i**, find the largest number which divides both **N** and **i** (that is, the greatest common divisor of **N** and **i**).

In order to unlock the cipher, you must find the sum of all the secret numbers. However, since this sum may be very large, you should output the remainder when this sum is divided by **M**.

To finally escape Phildystopia, you must find the sum of the secret numbers mod **M.**

Input Format

Two integers **N** and **M.**

Output Format

A single integer denoting the sum of the secret numbers modulo **M.**

Sample Input	**Sample Output**	**Explanation**
10 10000	27	1 + 2 + 1 + 2 + 5 + 2 + 1 + 2 + 1 + 10 = 27 27 (mod 10000) = 27
8 10	0	1 + 2 + 1 + 4 + 1 + 2 + 1 + 8 = 20 20 (mod 10) = 0