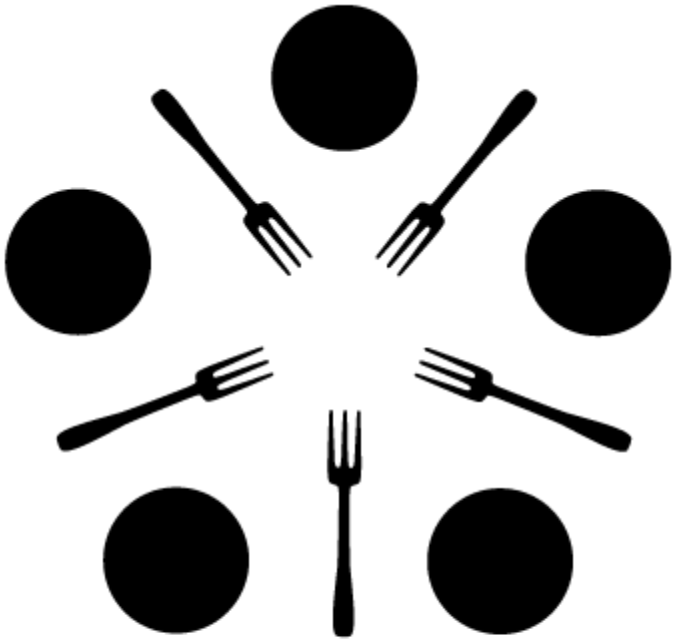
Contest Problems
Philadelphia Classic, Spring 2016
Hosted by the Dining Philosophers
University of Pennsylvania



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 10-14. These problems are correspondingly labeled “Novice”, “Intermediate”, and “Advanced.” Order does not otherwise relate to difficulty, except that problem 12 is the hardest.

You may use the Internet only for submitting your solutions, reading Javadocs, 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**. If you would like to solve a problem in Java, we have provided a stub file that takes care of the parsing for you. If you would like to solve a problem in Python, you must create a file for the answer and parse the input file yourself. It may be useful to refer to the Java stubs for how this works. Python submissions should be named the same thing as the Java stub, but with a .py extension instead of .java (“EscapeVelocity.py” instead of “EscapeVelocity.java”, for example). If you use Python, you may refer to the Python standard library docs.

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. Measuring Lightsaber Length

The Jedi Order has requested new lightsabers to be made for Jedi recruits in training. Fortunately, you and your team are very experienced in lightsaber construction. A proper lightsaber must be of appropriate length in order for the Jedis to properly use them. You know that the length of the lightsabers is based on the formula shown below:

$$Length = 2 \cdot P \cdot R$$

Note that P denotes the power in megawatts and R denotes the radius of the lightsaber in centimeters. The length of the lightsaber will be in meters.

You will be given two inputs of integer type. The first input will be the power in megawatts and the second will be the radius of the lightsaber in centimeters. Your job will be to compute the lightsaber length in meters based off of those two measurements.

Input	Output	Explanation
8 4	64	8 is the power, 4 is the radius of the lightsaber. The length according to the formula is 64.
8 1	16	8 is the power, 1 is the radius of the lightsaber. The length according to the formula is 16.
0 2	0	0 is the power, 2 is the radius of the lightsaber. The length according to the formula is 0.

2. Death Star Cipher Cracking

You are an expert in cryptography who has been hired by the rebel alliance to infiltrate the Death Star! Your job is to hack into Darth Vader's personal computer. Unfortunately, all the rebels know is that the password is somehow based on Vader's favorite foods, places, people, and things. However, a tip just came in informing you that the password is also somehow based on the vowels in those words. After hours of testing, you discovered that Vader makes his passwords a combination of the vowel that is closest to either the head or tail of the names of his favorite things.

You will be given a string containing at least one vowel. Your job is to return the vowel closest to either the head of tail (start or end) of the string. If there are two vowels equidistant from the ends of the string, return the vowel closer to the head. Return this letter as a string.

Note: the input string will only contain lower-case letters.

Input	Output	Explanation
pickle	e	"e" is the last vowel in the string and is the closest vowel to either side
indonesia	i	in the case that there are vowels at equal locations towards the ends of the string, the first vowel is returned

3. Message Origin Tracing

The Imperial Military has recently invented technology to intercept messages from the Rebel Forces. However, they need to know where the Rebels are sending these messages from in order to allow them to launch surprise attacks. You have been given the set of messages, each directly to an intermediate point of interception. However, being an expert in data tracing, you know that you can find the original point of transmission of each message by following these intermediate links back to where the message content and the point of interception are the same.

You will be given an array index and array of integers where each element is the index of that element's parent or its own index if it is the origin. For example, if an array at index 5 contains a 2, the element at index 2 is the parent, following this pattern until the element and index are the same, indicating the origin. You will find the origin of the element at the array index given. All arrays given are 0-indexed; in other words, the first number of each array has index 0.

Input	Output	Explanation
index: 6 array: 4 2 2 3 3 2 7 2	2	The array is [4, 2, 2, 3, 3, 2, 7, 2], and the start index is 6. At the index 6, 7 is the value of the array. You then go to index 7 where the value is 2, which traces to index 2 where the value is 2.
Index: 3 array: 4 2 2 3 3 2 7 2	3	At the index 3, the value in the array is also 3. Thus it is a root and returns 3.

4. The Battle of Jakku

The Imperial Military and the New Republic are at the final stage of the Battle of Jakku. You must help the Imperial Military determine if their weapons are strong enough. The Imperial Military has secretly and stealthily obtained the weapon inventories of the New Republic. However, the weapon information for each inventory is encoded in binary, and comes with a particular operation. You must perform this particular binary operation on all the encoded weapons to determine if the Imperial Military's weapons are strong enough. Can you help the Imperial Military?

You will be given two lists of strings such that each element in the list is a zero or a one. You will also be given a particular binary operation to apply to each corresponding element of the two lists. You must output the resulting list producing on applying the given binary operation to the the corresponding indexed elements of both the input lists.

Sample Input	**Sample Output**	**Explanation**
1 0 1 1 0 1 1 0 &	0 0 1 0	1 & 0 = 0, 0 & 1 = 0, 1 & 1 = 1, 1 & 0 = 0
1 1 ^	0	1 ^ 1 = 0 1 XOR 1 is 0
1 0 0 0 |	1 0	1 | 0 = 1, 0 | 0 = 0

5. Galactic Civil War

During the Galactic Civil War, the Rebel Alliance succeeded in stealing secret plans from the Galactic Empire's Death Star. To their dismay, it appears that the Empire has acquired a new superweapon capable of destroying entire worlds.

R2-D2 and C3-PO, in order to disable the superweapon, have landed on the Death Star. However, they have landed themselves in some serious trouble. They are being ambushed by stormtroopers! In order to escape, the droids must trick their adversaries by firing bullets that satisfy two conditions. The Droids can only fire bullets that are greater than a certain threshold value. Can you help the Droids escape the evil stormtroopers?

You will be given a list of numbers containing bullet values and a threshold value **K**, find the number of pairs of bullet values at indices (i,j) where **i < j** such that **abs(A[i] - A[j]) >= K.** In the above expression, **abs** denotes the absolute value of A[i] and A[j]. Note the fact that **i < j**

Input	Output	Explanation
K: 1 Numbers: [3, 1, 3]	2	(3,1) and (1,3) are the two pairs whose difference is greater than or equal to 1, based on the rules provided.
K: 3 Numbers: [1, 2, 3, 4]	1	The only pair is (1,4), whose difference is exactly 3.

6. Yoda Training

"Use more strength than you need you must not." - Yoda

A Jedi Master Yoda is. Grand Master of the Jedi Order in the waning days of the Galactic Republic he was. Lightsaber combat and the force he is renowned for.

A young padawan you are. A novice. Learning how to yield the powers of the Force, yes. Much Yoda has taught. Much you have learned. But know how to limit your strength you do not, hmm? Use more strength than you need you must not. When a ray gun is enough use a lightsaber you cannot.

Write a program to choose a weapon you must. A collection of weapons you will be given. The weakest weapon which is just strong enough you will determine.

Your program will be given the root node "root" of a balanced binary search tree of weapons and a minimum weapon strength "k". Each node in the tree represents a weapon. It will have a "strength" and a left child "left" and right child "right". The higher the "strength" value, the stronger the weapon is. Strength values can be negative.

Return the strength value of the weakest weapon in the tree which is strictly stronger than k. In other words, your program should return the value in the binary search tree which is closest to but still larger than k.

Note: There will always be a value in the tree which is larger than k.

Input Limits

There will be at most 100 test cases.

$1 <= |V| <= 100000$

$-2^{24} <= k <= 2^{24}$

Sample Input	**Sample Output**	**Explanation**
tree: 3 / \ 1 6 \ / \ 2 5 7 k: 2	3	3 is the smallest value in the tree which is larger than 2.
tree: 3 / \ 1 6 \ / \ 2 5 7 k: 1	2	2 is the smallest value in the tree which is larger than 1.

7. Pure Pazaak

You find yourself stranded on Tatooine without enough credits to buy fuel for you ship. Being an industrious traveler you decide to make the credits you need by winning them in a card game called Pazaak. You head to Mos Eisley and find the nearest gambling den where patrons are playing this traditional card game. This two-player game starts with a series of cards, each with an integer value. On each turn, a player selects either the first or the last card from the sequence and adds it to their pile. The game ends when there are no cards left to choose, and the winner is the player with the largest sum of cards in their pile. Since you can't afford to lose any games you decide to write a program to determine what move you should make at every turn.

You will be given as input a list of integers of even size representing the values of the cards at the start of the game. You will return the sum of the cards you will choose if you play with your algorithm. This score should assume that you play first and your opponent also plays optimally. Note that this means if your algorithm determines the optimal play then your opponent will also be playing with this strategy.

Input limits

The number of cards will less than 10,000.

Sample Input	**Sample Output**	**Explanation**
[4, 5]	5	Clearly the best thing to do here is just pick the largest value
[10, 100, 4, 1]	101	You pick 1 and then no matter what your opponent picks you pick 100
[8, 3, 10, 5, 7, 2]	25	You choose 8. If your opponent chooses 2 you choose 7 then 10 and if your opponent chooses 3 you choose 10 then 7.

8. Droid Quality Assurance

Unkar Plutt, a member of the male Crolute species, is a junkboss who buys and sells weapons, gears, and droids on the planet Jakku. The ruthless Unkar employs a band of scavengers who comb the desert looking for junk.

Unkar notices that the most common defect in a droid is if it has been miswired to short circuit. A droid will short circuit if their circuit board contains a sequence of pins which form a loop. Since a short-circuiting droid is almost worthless, Unkar needs a program to quickly determine if a droid will short circuit (so he can avoid buying these droids).

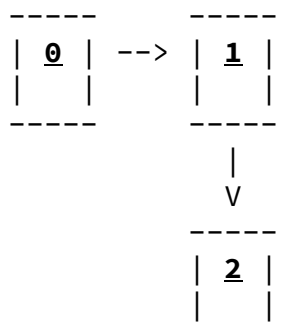
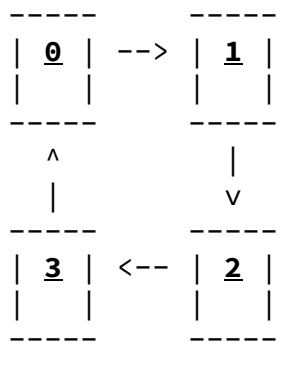
Your program will be given a collection of pins on a droid's circuit board. Each pin will have a list of other pins called links, which are all of the other pins a pin connects to. It should return whether or not the collection of pins contains a short circuit.

Note: The collection of pins and connections is *directed*. Pin 1 can have a connection to pin 2 without pin 2 having a connection to pin 1.

Input Limits

There will be at most 20 test cases.

1 <= |V| <= 10,000

Sample Input	**Sample Output**	**Explanation**
	false	This circuit board does not contain a short circuit
	true	This circuit board does contain a short circuit

9. Death Star Construction

You are a member of the Imperial Fleet assigned to aid in the construction of what seems like the 10th Death Star or Death Star like space station. Specifically you are supposed to use a crane to move the large plasma cutting beams out of the construction area.

Some genius officer decided that even though in the void of space you could move the beams in any direction you would be given a crane that can only move the beams in one direction. For efficiency reasons you would like to use one motion to lift each beam (that is each time you pick up a beam you remove it from the construction area). However since the volatile nature of the beams means they can't touch the one dimensional nature of your crane means there are some arrangements of beams that you can't easily remove one beam at a time. You would like to write a program to determine if a given set of beams can be easily removed from the construction area so you can call in sick if they can't.

You will be given a set of beams in the form of pairs of ordered triplets in the form (x, y, z). These triplets define two points in space which in turn define the endpoints of each beam. Your crane can only lift in the positive "z" direction. You will return true if the beams can be easily removed and false if they cannot.

Input limits

Each test case will contain at most 1,000 beams.

Input	**Output**	**Explanation**
{(-1 -1 0) (1 1 0)} {(-1 -1 1) (1 -1 1)}	true	There are two beams, one from {-1, -1, 0} to {1, 1, 0} and one from {-1, 1, 1} to {1, -1, 1}. This represents two sticks which are crossed on on top of each other so you can just lift the top one and then the next one.
{(-1 2 -1) (2 -2 3)} {(0 2 2) (0 -1 -1)} {(2 0 -1) (-1 0 2)}	false	These beams are arranged in an interlocking triangle so trying to lift anyone results in you hitting the end of a different beam. Call in sick, this is not going to be fun.
{(-1 2 -1) (2 -2 2)} {(0 2 2) (0 1 1)} {(2 0 -1) (-1 0 2)}	true	This is the same situation as before but now one of the beams is short enough that you can lift it up without hittin the other two.

10. Fluctuations in the Force

The Jedi Council is trying to seek out potential traces of Sith Lords by tracking fluctuations in the Force. As you know, the Light and Dark sides of the Force are in constant flux, but the Jedi council believes that there are patterns that may indicate the malevolent influence of Sith Lords.

To this end, they have been able to measure the balance over a number of days, which is summarized as a series of numbers (higher numbers represent a stronger Light side, lower numbers represent a stronger Dark side). The patterns that the Jedi council believe are malevolent are triples of days (not necessarily next to each other) where the balance is continually decreasing. For example, if we have measured the numbers [4, 5, 2, 1] over 4 days, then days 1, 3 and 4 would form such a malevolent pattern, since the values [4, 2, 1] are strictly decreasing. Thus, given the measurements over a sequence of days, help the Jedi council count the number of malevolent pattern!

Note: Your program will not run fast enough if you try all possible triples in nested for loops! It will not run fast enough even if you only try all possible doubles in nested for loops!

Input limits

The maximum length of a sequence of days will be 100,000.

Sample Input	**Sample Output**	**Explanation**
[4, 5, 2, 1]	2	The triples (4, 2, 1) and (5, 2, 1) are malevolent.
[6, 1, 2, 4, 5, 3]	2	The triples (6, 5, 3) and (6, 4, 3) are malevolent.
[5, 4, 3, 2, 1]	10	All triples are malevolent. Better alert the Council!

11. Black Market Mania

You are fighting in the Clone Wars on the planet Mandalore. As the planet has been cut off from the Galactic Republic, Almec has established a black market for essential goods. In particular, he has set up a market in which there are n goods, and each good can be exchanged for some quantity of any other good. You start out with 10 credits (which can be used to buy the first good on the black market). You'd like to write a program to determine if it's possible to start with this 10 credits, buy and sell goods repeatedly, and end with more than 10 credits. That is you would like to determine if there is some sequence of trades that leaves you with more than you started with.

You will be given as input a matrix of doubles where the (i,j) entry represents how many units of good i you can trade for a single unit of good j. In other words, the row i gives the purchase price of all other goods in terms of i. For all i, the (i,i) entry is 1 (i.e. goods cannot be bought and sold for themselves). You must output a boolean indicating whether it is possible to create a profit simply by buying and selling goods, starting with 10 credit (true if possible, false if impossible).

Input limits

There will be at most 300 goods on the market.

Input	Output	Explanation
[[1.0,2.0], [1.0,1.0]]	false	You can purchase up to 5 units of good 2 with your 10 credit, but if you then try to re-purchase credit, you only end with 5.
[[1.0,1.0], [0.5,1.0]]	true	You can purchase up to 10 units of good 2 with your 10 credit, and then you can re-purchase 20 credit, yielding a profit.

12. Jedi Academy Squadrons

You are the member of the Council of First Knowledge and you've recently been promoted to run your own Jedi Academy on Coruscant. As your first goal, you've decided to split your students up into **g** different training groups. You want everyone in a training group to have a similar final exam score so you come up with the following metric to measure how good a particular arrangements of groups: For a specific group G we let the "cost" of that group to be the sum of the squared differences between each group member's score and the average score for that group. We then let the total cost for an grouping to be the sum of the costs of each group. You would like to write a program to compute the optimal (i.e. minimum cost) grouping of the students into **g** training groups.

Input limits

The number of students in each test case will be less than 3,000. The number of groups we wish to divide the students into will be less than 50.

Input	Output	Explanation
g = 4 [1,4,3,2]	0	Each student can be in their own group, allowing for zero total cost
g = 2 [3,1,2,4]	1	With 2 groups, we can put the students with scores 1 and 2 in one group, and with 3 and 4 in the other group. The average of the first group is 1.5 and the average of the second group is 3.5, so the score is (1.5 - 1)^2 + (1.5 - 2)^2 + (3.5 - 3)^3 + (3.5 - 4)^2 = 1