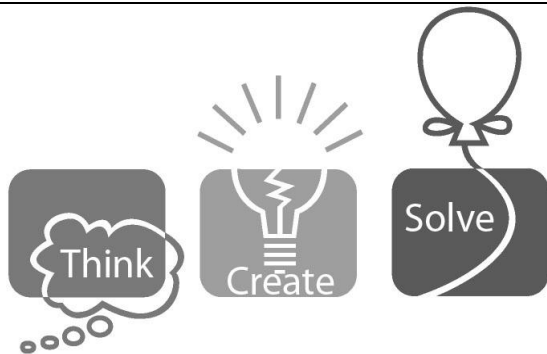


University of Central Florida



2026 (Fall)
“Practice/Qualifying” Local
Programming Contest

Problems

Problem#	Difficulty Level	Filename	Problem Name
1	Easy	bills	Briefcases Full of Money
2	Easy	mind	A Game Called Mind
3	Easy	digit	Digit Count
4	Easy-Medium	box	Tetrooj Box
5	Easy-Medium	autojudge	Judging Assistant for Contest
6	Medium	connect	Making Connections
7	Medium	badtree	Bad Tree
8	Medium	unique	Unique Values
9	Medium	sumfunc	Sum of a Function
10	Medium-Hard	food	Food Display Arrangement
11	Medium-Hard	hangglide	Hang Gliding
12	Medium-Hard	drink	Thirsty Professors
13	Medium-Hard	fish	Gone Fishing

Call your program file: filename.c, filename.cpp, filename.java, or filename.py

For example, if you are solving Briefcases Full of Money:

Call your program file: bills.c, bills.cpp, bills.java, or bills.py

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Briefcases Full of Money

filename: bills

Difficulty Level: Easy

Time Limit: 5 seconds

It is your birthday party and the six UCF programming team coaches arrive, each holding a briefcase containing money (gift) for you. The coaches were planning to give you the six briefcases but Dr. “O” points out that the team needs money to travel to World Contest Finals! So, you get to choose one briefcase, i.e., you are not getting all the briefcases (sorry).

Each briefcase contains a stack of bills; each briefcase containing one of the 6 denominations \$1, \$5, \$10, \$20, \$50, \$100, i.e., first briefcase contains only \$1 bills, second contains only \$5 bills, third only \$10 bills, fourth only \$20 bills, fifth only \$50 bills, and sixth only \$100 bills. You, of course, want to pick the one with the highest total amount.

The Problem:

You can randomly pick one briefcase but, as a programmer, you trust your coding skills more than chance and decide to write a program to help you pick the briefcase with the highest amount.

The Input:

There is only one input line; it contains six integers (each integer between 1 and 10^3 , inclusive). These integers represent, respectively, the number of \$1, \$5, \$10, \$20, \$50, \$100 bills.

The Output:

Output which briefcase to choose by printing the denomination in that briefcase (1, 5, 10, 20, 50, 100). If two or more briefcases have the highest total, pick (print) the one with fewest number of bills among those briefcases since that one is lighter!

Sample Input

Sample Output

84 111 2 3 2 3	5
200 3 20 5 4 1	50
1000 2 2 2 2 2	1

Explanation of the second Sample Input/Output: Three briefcases (\$1, \$10, \$50) have the highest total (\$200) so the output is the briefcase with the fewest number of bills.

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

A Game Called Mind

filename: mind

Difficulty Level: Easy

Time Limit: 5 seconds

This problem deals with a simplified version of a game called “Mind”. In this game, a group tries to cooperatively solve a problem. Assume:

- There are 2-6 players (called A, B, C, D, E, F).
- Each player has 1-9 cards in sorted order.
- Each card value is between 10 and 99 (inclusive).
- There are no duplicate values, i.e., a card (number) is in at most one hand.

The Problem:

The objective of the game is for all the players to put all the cards down in sorted order (in the real game, players do not see their own cards, hence the name for the game – Mind). Given the cards in each hand, you are to provide the order for the players to place their cards down.

The Input:

The first input line contains an integer, p ($2 \leq p \leq 6$), indicating the number of players. Each of the following p input lines provides the cards for one player. Each line starts with an integer, c ($1 \leq c \leq 9$), indicating the number of cards for that player; the count is followed by the cards in sorted order (each card between 10 and 99, inclusive).

The Output:

Print the order for the players to place down the cards so that the cards are in order.

Sample Input

Sample Output

2 3 10 40 50 2 20 30	ABBAA
3 4 40 51 60 70 3 12 32 42 5 20 53 80 90 95	BCBABACAACCC

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Digit Count

filename: digit

Difficulty Level: Easy

Time Limit: 5 seconds

There are many ways to count the frequencies of letters but can they be applied to digits?

The Problem:

Given a range (in the form of two integers) and a digit (0-9), you are to count how many occurrences of the digit there are in the given range.

The Input:

There is only one input line; it provides the range and the digit. Each integer for the range will be between 1000 and 9999 (inclusive) and the digit will be between 0 and 9 (inclusive). Assume the first integer for the range is not greater than the second integer for the range.

The Output:

Print the number of occurrences of the digit in the given range.

Sample Input

Sample Output

1000 1000 0	3
1000 1001 0	5
8996 9004 5	0
9800 9900 5	20

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Tetrooj Box

filename: box

Difficulty Level: Easy-Medium

Time Limit: 5 seconds

Dr. Orooji’s children have played Tetris but are not willing to help Dr. O with a related problem. Dr. O’s children don’t realize that Dr. O is lucky to have access to 100+ great problem solvers and great programmers today!

The Problem:

Dr. O knows the length of the base for a 2D box and wants to figure out the needed height for the box. Dr. O will drop some 2D blocks (rectangles) on the base. A block will go down until it lands on the base or is stopped by an already-dropped block (i.e., it lands on that block). After all the blocks have been dropped, we can determine the needed height for the box – the tallest column is the needed height (please see pictures on the next page corresponding to Sample Input/Output).

The Input:

The first input line contains two integers: b ($1 \leq b \leq 100$), indicating the length of the base and r ($1 \leq r \leq 50$), indicating the number of blocks (rectangular pieces) to be dropped. Each of the next r input lines contains three integers: a block’s horizontal length h ($1 \leq h \leq 100$), the block’s vertical length v ($1 \leq v \leq 100$), and c ($c \geq 1$), the leftmost column the block is dropped into. Assume that the h and c values will be such that the block will not go beyond the box base, i.e., $(c + h - 1) \leq b$.

The Output:

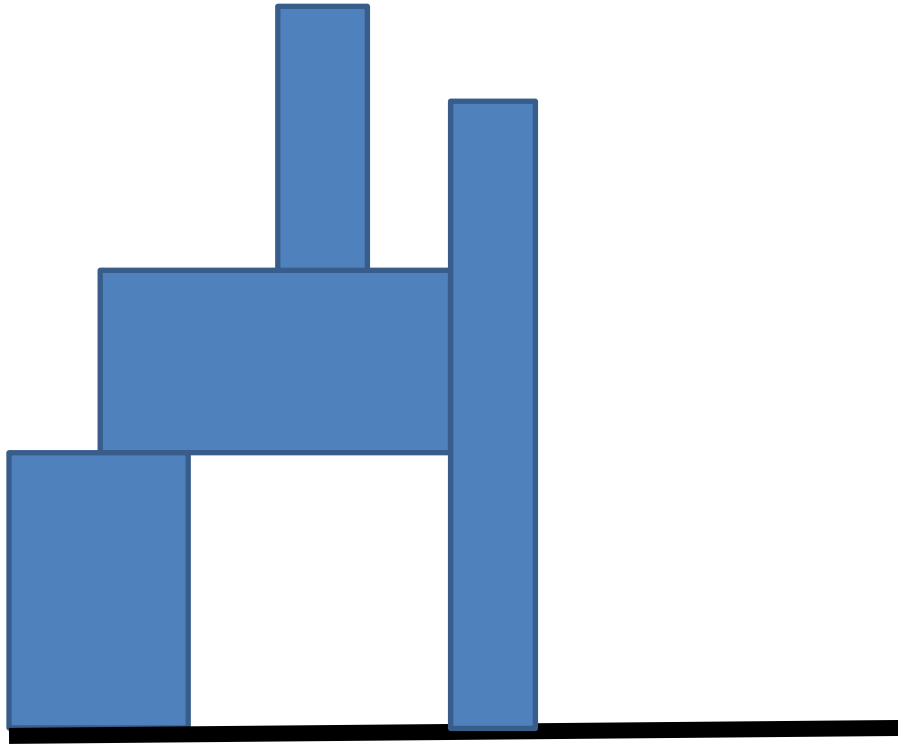
Print the needed height for the box (the tallest column is the height).

Sample Input

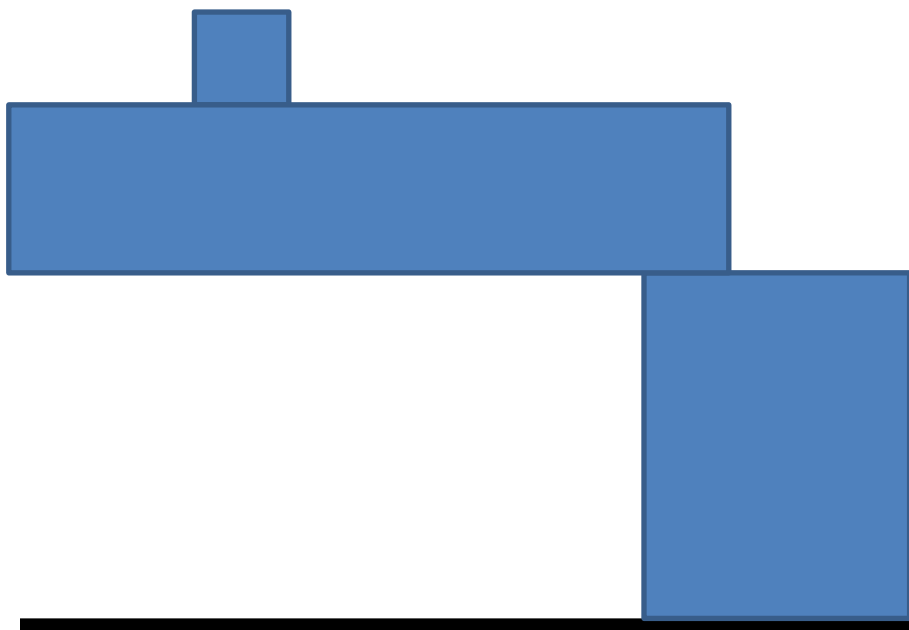
Sample Output

10 4 2 3 1 4 2 2 1 7 6 1 3 4	8
10 3 3 4 8 8 2 1 1 1 3	7

First Sample Input/Output:



Second Sample Input/Output:



UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Judging Assistant for Contest

filename: autojudge

Difficulty Level: Easy-Medium

Time Limit: 5 seconds

When people try competitive programming for the first time, it can be challenging for some to write code according to the constraints of the contest, even if they are already good coders! Of course, the best way to learn is by doing, and that is why many contests have a “practice” or warm-up session, which includes testing out the full process of submitting code and getting responses from the judges. Still, there are always some people who do not take full advantage of that session and then make time-wasting mistakes in the real contest, where it counts! For example, they print out prompts for inputs, or forget to return zero from their C program. This happens often enough that the judges need some programs to help them evaluate the submissions.

The Problem:

Given information about a programming contest problem and a submission for that problem, help the judges determine the best response.

The Input:

The first input line contains the designated “filename” for the contest problem which is a string of 1 to 20 lowercase letters. Recall that this filename does not include the extension (.c, .cpp, .java, .py).

The second input line contains the name of the submitted file, a string of 1 to 70 characters. This filename may include an extension, though the contestant might have used an invalid extension (e.g., .html, .pl, .rb, .asp, ...). Note that this file name consists of characters and not just letters, e.g., the file name may be “C:\My Documents\graph.py”.

The third input line contains three single-space-separated integers: r ($0 \leq r \leq 10$), indicating the return code of the submitted program, d ($1 \leq d \leq 10$), indicating the time limit in seconds allowed for a correct program to run, and e ($0 \leq e \leq 20$), indicating the elapsed time in seconds while the program was running. (Note that, in a real contest, it may not be possible to run a submitted program but that aspect is not included in this problem to simplify the problem.)

The fourth input line contains an integer, c ($1 \leq c \leq 10$), indicating the number of output lines produced by a correct program. The following c input lines provide the correct output; each line will contain zero to 70 characters and the first and last line(s) will contain at least one non-blank character.

The next input line contains an integer, t ($0 \leq t \leq 10$), indicating the number of output lines produced by the submitted program. The following t input lines provide the output produced by the submitted program; each line will contain zero to 70 characters. Note that, unlike the correct

output, it is not guaranteed that the first and last line(s) of the submitted output will contain at least one non-blank character, i.e., the submitted output could be all blanks. Note also that, in a real contest, a submitted program may have far more output lines than expected and some output lines may far exceed the expected length but those aspects are not included in this problem to simplify the problem.

The Output:

If the submitted file name does not match (case-sensitively) the designated problem filename, or if it doesn't have one of the valid extensions (.c, .cpp, .java, or .py), print the message "Compile Error". Otherwise, if the return code of the program is not zero, print "Run-Time Error". Otherwise, if the elapsed time for the submitted program exceeds the time limit, print "Time Limit Exceeded". Otherwise, if the submitted program output does not match the correct output (line by line and character by character), print "Wrong Answer". Otherwise, print "Correct".

Sample Input

Sample Output

triangle MyTriangle.py 0 3 0 1 scalene 0	Compile Error
graph graph.py 5 3 3 1 done 1 done	Run-Time Error
dust dust.c 0 10 11 3 12cm of dust 3cm of dust Impossible 1 Please enter shelf size:	Time Limit Exceeded

(more Sample Input/Output on the next page)

Sample Input**Sample Output**

<pre>awesome awesome.java 0 5 5 1 Result = 100 2 Result = 100</pre>	Wrong Answer
<pre>awesome awesome.java 0 5 5 2 Result = 100 2 RESULT = 100</pre>	Wrong Answer
<pre>awesome awesome.java 0 5 5 1 Everything is awesome! 1 Everything is awesome!</pre>	Correct

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Making Connections

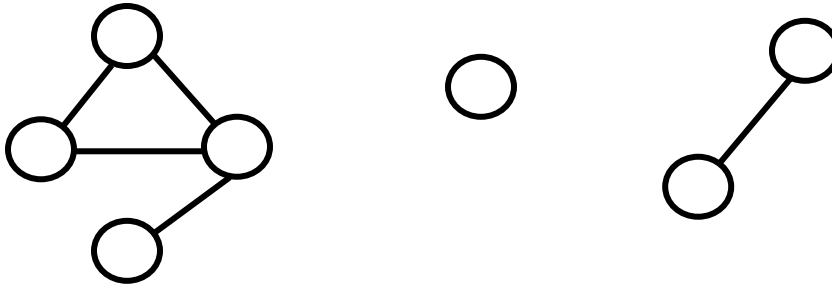
filename: connect

Difficulty Level: Medium

Time Limit: 5 seconds

Given that everything is online these days, connectivity is a must. A computer network can be modeled as a graph, where each computer is a vertex and each direct network connection between pairs of computers is an undirected edge.

Consider the process of building a computer network. At the very beginning there will be n computers, with no connections between any of them. Then, as time goes on, pairs of computers are chosen, one pair at a time, and a direct network connection between them is added. In the middle of such a process, we might get the following graph modeling the current connections:



This network currently has three groups: the first group has 4 computers that can communicate with each other directly or indirectly, the second group consists of 1 computer by itself, and the third group has 2 computers that can communicate with each other. So, a group is each of the separate components of the graph.

We can define the average connectivity of a network as the sum of the group sizes squared, divided by the number of components. For the example graph shown above, the current connectivity equals $(4^2 + 1^2 + 2^2)/3 = 21/3 = 7$.

As a network is being built, the project manager would like to know the average connectivity of the network at that given snapshot of time. Write a program to handle the queries as the network is being constructed.

The Problem:

Given a network of n initially separate computers, along with a sequence of steps, where each step is either a pair of computers being connected or a query about the average connectivity as of that moment, process each step (i.e., connect the two computers or answer the query).

The Input:

The first input line contains two space separated integers: n ($1 \leq n \leq 10^5$), representing the number of computers, and m ($1 \leq m \leq 3 \times 10^5$), representing the total number of steps (each step being either

connect two computers or a query on the average connectivity as of that moment). Assume that the computers are numbered 1 through n , inclusive.

Each of the following m input lines contains information about one operation, in the order that they occur. Each of these lines will start with a single integer, either 1 or 2, providing the step type. If the step type is 1, it means that a pair of computers is being connected with a direct connection. The value of 1 will be followed by u and v ($1 \leq u, v \leq n, u \neq v$), representing the pair of computers being connected with a direct connection. If the step type is 2, this is a query and no other information will be on that input line.

Note: It's possible that the input contains multiple direct connections for the same pair of computers; the extra direct connections between the same two computers do not have any effects. It's also possible that, at the end of the process, not all n computers are connected in the same component, i.e., there may be more than one component even at the end of the process.

The Output:

For each query, output the average connectivity of the network at that point in time as a fraction in lowest terms on a line by itself. Specifically, output two integers, x and y , with the character '/' in between, indicating that the average connectivity of the network at the time is x divided by y such that x and y share no common factors, e.g., 21/12 is not correct (should be 7/4).

Sample Input

Sample Output

7 9	1/1
2	13/5
1 1 2	19/4
1 1 3	7/1
2	
1 3 4	
1 2 3	
2	
1 6 7	
2	
4 9	2/1
1 1 2	4/1
2	16/1
1 3 4	16/1
2	
1 2 3	
2	
1 1 4	
1 2 4	
2	

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Bad Tree

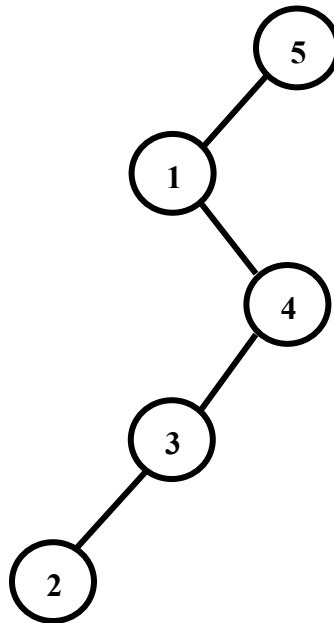
filename: badtree

Difficulty Level: Medium

Time Limit: 1 second

Binary Search Trees are supposed to speed up searches for items but this happens only when the height of the tree is much less than the total number of values stored in the tree. And, in programming contests, judges unfortunately try to make the “worst case data”. So invariably, in binary search tree problems, judges will make data where n nodes get inserted into a tree in a particular order so that the height of the tree is the worst case, $n - 1$.

Without loss of generality, let’s assume that the n items to be inserted into a binary search tree are $1, 2, 3, \dots, n$. For $n = 5$, if we insert the values in this order: 5, 1, 4, 3, and 2, we get the following binary search tree of height 4:



In fact, there are quite a few orderings of the first n positive integers that, when inserted into a binary search tree, create a binary search tree of height $n - 1$. Since you aspire to be a great judge one day for this contest, write a program to generate the k^{th} lexicographical permutation of the first n positive integers that, when inserted into a binary search tree in that order, generates a binary search tree of height $n - 1$.

The Problem:

Given a positive integer n , and another positive integer k , determine the k^{th} permutation, in lexicographical ordering, that when the items are inserted into a binary search tree in that order, generates a tree of height $n - 1$.

The Input:

There is only one input line; it contains two space separated integers: n ($1 \leq n \leq 100$), representing the number of nodes in the binary search tree, and k ($1 \leq k \leq 10^{18}$), where we desire the k^{th} lexicographical permutation of the first n integers which creates a binary search tree of height $n - 1$, when inserted in the order given in the permutation.

The Output:

Print the k^{th} lexicographical permutation of the integers 1 through n of the permutations which, when the values are inserted into a binary search tree, create a tree of height $n - 1$. Output the permutation on a single line, following each number in the permutation with a space. If no such permutation exists, output -1 instead.

Sample Input**Sample Output**

5 12	5 1 4 3 2
4 2	1 2 4 3
6 1	1 2 3 4 5 6
3 50	-1

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Unique Values

filename: unique

Difficulty Level: Medium

Time Limit: 2 seconds

Arup has to make many practice questions for his Computer Science 1 students. Since many of the questions deal with arrays, he has to generate arrays for his students. Since he doesn't want to give them difficult practice problems, he always guarantees that the arrays (given to the students) have unique values. Namely, no value will appear twice in any of his arrays.

Unfortunately, Arup is too lazy to generate arrays! About 25 years ago when he started teaching Computer Science 1, he made one really long array to reuse but this long array may have duplicate values. When he makes problems, he simply grabs a contiguous subsequence of this long array to be the array to be used for a problem but he needs to make sure the contiguous subsequence does not contain duplicates. If the long array has terms $a[0]$, $a[1]$, ..., $a[n-1]$, a contiguous subsequence of the long array is any sequence of $j - i + 1$ terms $a[i]$, $a[i+1]$, ..., $a[j]$ where $0 \leq i \leq j \leq n - 1$.

The Problem:

Given an array of n integers, determine how many contiguous subsequences of the array do not contain any repeated values. Note that two subsequences with the same contents are considered different (i.e., both counted in the total) if they start at different locations of the original long array.

The Input:

The first input line contains a single positive integer, n ($1 \leq n \leq 10^5$), representing the size of the input array. The following line contains n space separated integers, representing the values of the input array, in the order they appear in the array. Each of the array values will be an integer between 1 and 10^9 , inclusive.

The Output:

On a line by itself, output the total number of contiguous subsequences of the input array that do not contain any repeated values.

Sample Input

Sample Output

5 1 1 2 1 5	9
8 2 12 3 12 3 2 6 9	22

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Sum of a Function

filename: sumfunc
Difficulty Level: Medium
Time Limit: 1 second

Everyone knows that Arup loves prime numbers! This is why he teaches the cryptography course at UCF. Recently, Arup defined the following function on positive integers, n , greater than 1:

$f(n)$ = the smallest prime factor of n

For example, $f(14) = 2$, $f(15) = 3$, $f(16) = 2$ and $f(17) = 17$.

Using this function, we can generate a sequence of terms $f(s)$, $f(s+1)$, $f(s+2)$, ..., $f(e)$, where s designates the starting function input and e designates the ending function input.

Arup thinks these sequences are interesting, but what he’s really curious about is finding the sum of the k minimum elements in one of these sequences. Can you write a program to help him?

The Problem:

Given s , e , and k , find the sum of the k minimum values in the sequence $f(s)$, $f(s+1)$, $f(s+2)$, ..., $f(e)$.

The Input:

The first and only input line will contain three positive integers, s ($2 \leq s \leq 10^{18}$), e ($s+100 \leq e \leq s+10^6$), and k ($1 \leq k \leq 0.9 * (e - s + 1)$), representing (respectively) the starting function input, the ending function input, and the number of minimum terms to sum.

The Output:

On a line by itself, print the sum of the k minimum terms of the designated sequence.

Sample Input

Sample Output

100 200 70	165
213 419 169	546

Note: Even though the input specification does not allow “14 17 3” as an input case (i.e., this case will not be in the judge data), it is a simple case that you may want to use for testing purposes – the output (7) can be verified easily on paper. (BTW, the intended solution should solve this case properly anyway.)

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Food Display Arrangement

filename: food

Difficulty Level: Medium-Hard

Time Limit: 10 seconds

Your friend, Thomas, is working on Food Display Arrangements (FDA). He has all the food lined up on a long row (table). His job requires that he arranges the FDA in an aesthetically pleasing manner. An FDA is aesthetically pleasing if all the food of the same type is grouped together, i.e., all the food of the same type are next to each other. Thomas can reorganize the FDA as follows: pick up all the food of one type and place it on either end of the table, i.e., place it at the beginning of the table or at the end of the table. Thomas wants to know the minimum number of reorganization steps needed to make the FDA aesthetically pleasing. Note that you don’t need to tell him the specific steps, only the least number of steps.

The Problem:

Given a display of food by their types, determine the minimum number of times necessary to move all food of the same type to the end or the beginning of the display to ensure that all food of the same type is grouped together. Assume that the display can be extended at the ends to contain any amount of moved food.

The Input:

The first input line contains an integer, n ($1 \leq n \leq 100,000$), representing the number of food items in the display. The next input line contains n space separated integers, a_i ($1 \leq a_i \leq 1,000,000,000$), representing the id of the i^{th} food item in the display.

The Output:

Print the minimum number of times necessary to move all food of the same type to the beginning or the end to make the FDA aesthetically pleasing.

Sample Input

Sample Output

15 8 8 2 8 7 8 1 1 3 7 3 4 2 4 7	2
10 1 2 3 4 5 1 2 3 4 5	4

Explanation of the first Sample Input/Output: We can move all food of type 2 to either end and all food of type 7 to either end, for a total of 2 moves.

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Hang Gliding

filename: hangglide

Difficulty Level: Medium-Hard

Time Limit: 3 seconds

The 2027 hang gliding world championships are coming to Florida next spring! (You may think it is odd to hold in Florida a sport that requires mountains, but it turns out that hang gliders can be towed to altitude by other aircraft.) The competition is divided into a set of tasks; completing a task successfully gives a pilot a number of points. Either all points are awarded for a task, or none are. For each task, every pilot has a probability of success. A pilot’s score is the total of all successfully completed tasks at the end of the competition.

This year, the organizing committee couldn’t decide on a reasonable number of tasks, so the time slots for tasks overlap. At any given time, there can be multiple tasks going on, but a pilot may only choose one to be competing in at that time. Each pilot may compete in as many tasks as they want given this constraint. The pilots know their own strengths and weaknesses, and will choose tasks in order to maximize their expected score.

You have been offered free hang gliding lessons if you help with scoring software for the event. Among other things, the committee wants the software to be able to predict the winners ahead of time.

The Problem:

Given a set of tasks, each with a time slot and a point score to be awarded for success, and a list of pilots, each with success probabilities for each task, compute the expected score for each pilot, and report the top 3 expected scores.

The Input:

The first input line contains two integers: t ($1 \leq t \leq 10^4$), indicating the number of tasks, and p ($3 \leq p \leq 100$), indicating the number of pilots.

The next t input lines contain the task descriptions. Each line contains three integers (s , e , and a) separated by a space: $0 \leq s < e \leq 10^4$, s is the start time of the task, and e is the end time of the task, in minutes after the competition starts; a ($1 \leq a \leq 100$) is the number of points awarded for the task. Note that the task start times and end times are non-inclusive, i.e., if a task ends at time T and another task starts at time T , a pilot can compete in both tasks.

After the task descriptions, there are t lines for each pilot. The first t lines in this section are the probabilities of success for each task for pilot 1; the next t lines are the probabilities of success for pilot 2, and so on. The probabilities are floating point numbers in the range 0 to 1, inclusive.

The Output:

Print the top 3 pilots. Print, in order of descending expected score, the pilot's number, a space, and the pilot's expected score, rounded to 2 decimal places (i.e., a score of 42.494 would be printed as 42.49; a score of 42.495 would be printed as 42.50). Note that pilots are numbered starting at 1, not zero. Assume that there will not be any ties for the top 3, i.e., there will not be multiple correct answers.

Sample Input

Sample Output

3 3 0 1 5 1 2 10 2 3 15 0.0 0.0 0.2 1.0 0.0 0.0 0.0 0.75 0.0	3 7.50 2 5.00 1 3.00
3 4 0 3 50 3 6 50 0 6 75 0.2 0.3 0.6 0.6 0.6 0.5 0.6 0.5 0.4 0.9 0.9 0.9	4 90.00 2 60.00 3 55.00

UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Thirsty Professors

filename: drink

Difficulty Level: Medium-Hard

Time Limit: 3 seconds

Dr. Orooji and Dr. Mentor were lost in a desert and they were extremely thirsty. They each had a stick so they decided that they can form a shape (e.g., “V” shape, “X” shape, etc.) facing the skies and, when it rains, water would collect in the top part of the shape and then they can drink it; please see the picture on the next page.

Making the shape would be easier if one person holds both sticks but neither professor was willing to give up his stick. So, they tried to form the shape together, each holding one stick. And, we know how coordinated the professors can be!

The Problem:

Given two line segments, the first line with positive slope and the second line with negative slope, compute the area for water collection. When checking for intersecting (touching), if two points are within 10^{-6} of each other, consider the points the same.

The Input:

There are two input lines, each describing a line segment. The first input line contains four integers ($0 < x_1, y_1, x_2, y_2 < 10^3$; $x_2 > x_1$ and $y_2 > y_1$), describing the first stick. The second input line contains four integers ($0 < x_3, y_3, x_4, y_4 < 10^3$; $x_4 < x_3$ and $y_4 > y_3$), describing the second stick. Note that neither line segment will be parallel to x -axis or parallel to y -axis.

The Output:

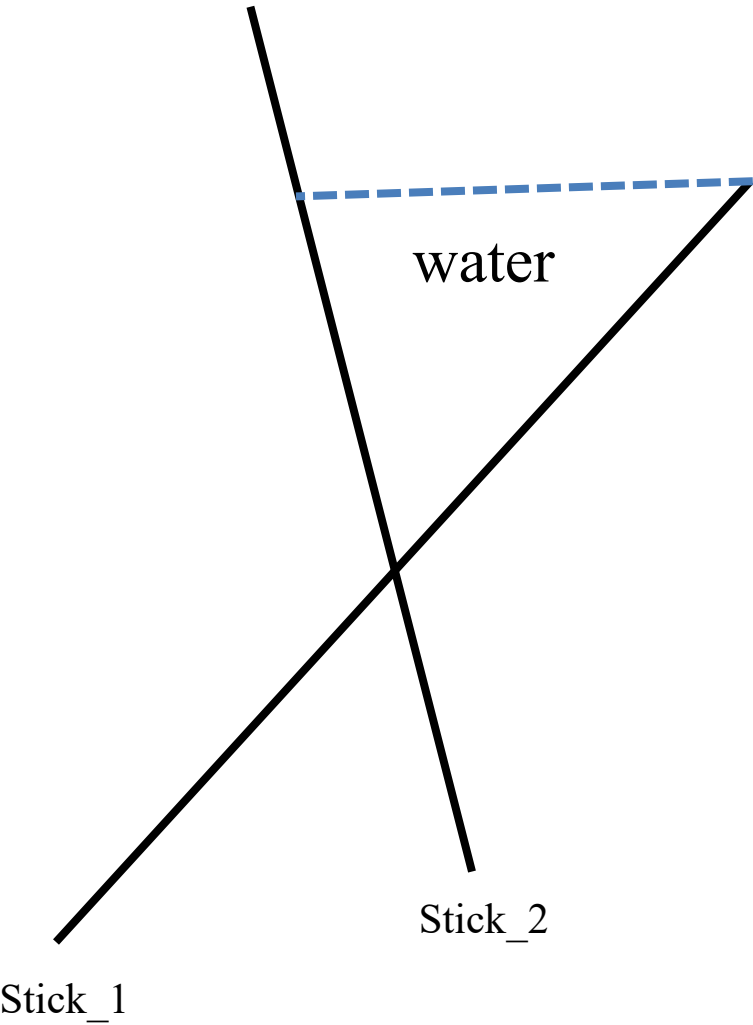
Print the area for water collection. Your output may have any number of digits after the decimal point; answers within 10^{-5} absolute or relative of the judge output will be considered correct.

Note that, as illustrated in the second Sample Input/Output, if the two line segments do not intersect, it will not be possible for the water to collect and, as such, the output will be zero.

Sample Input

Sample Output

5 2 9 6 5 2 3 4	4.0
11 14 18 15 13 18 10 20	0.0
5 2 9 6 12 1 6 10	0.03333



UCF “Practice/Qualifying” Local Contest — Sept 1-7, 2026

Gone Fishing

filename: fish

Difficulty Level: Medium-Hard

Time Limit: 1 second

It is getting dark and the mosquitoes are attacking Fisherman Floyd. Floyd decides to throw his circular net one last time and wants to make the most out of the last throw.

The Problem:

Given the size (radius) of Floyd’s net and positions (x,y) of a set of fish, what is the maximum fish Floyd can catch with one throw? That is, find the maximum points you can have in the net (circle). If a point (fish) is within 10^{-6} of the circle boundary, consider the point in the circle.

The Input:

The first input line provides the radius of the circle. The second input line contains an integer, n ($1 \leq n \leq 100$), indicating the number of fish. Each of the next n input lines provides the location (x,y) for one fish. Assume all these points are distinct. Also assume that all the x and y values in the input (as well as the circle radius) are integers between 1 and 10^3 , inclusive.

The Output:

Print the maximum fish Floyd can catch with one throw.

Sample Input

Sample Output

20 4 6 7 5 5 190 100 4 4	3
3 2 1 1 95 4	1