2015 Australian and New Zealand Algorithmic Competition

Round 2

April 18

Start Time 1:30 (AEST)

End Time 5:30 (AEST)

Instructions

- For all problems, read the input from standard input, and write results to standard output.
- The time limit for all problems is 5 seconds.
- Don't use the Internet and teams should have at most 3 people.
- Solutions may be submitted in Java, Python 3, C#, C or C++. There is no guarantee that all problems will be solvable within the time limit in all of the allowable programming languages.

Problems collated (but not authored) by the organisers of this round.

Andrew Haigh and Eric McCreath

Problem sources

Name	Source
Yuangfang, what do you think?	Jinhua 2012
Wally World	East Central 2011
Stealing Harry Potter's precious	Hangzhou 2013
Please, go first	NWERC 2011
Kickdown	NEERC 2006
iSharp	NEERC 2008
Interconnect	NEERC 2006
Grachten	NWERC 2013
Game	NEERC 2007
Fibonacci	NEERC 2008
Asteroids	NEERC 2009

Asteroids

Association of Collision Management (ACM) is planning to perform the controlled collision of two asteroids. The asteroids will be slowly brought together and collided at negligible speed. ACM expects asteroids to get attached to each other and form a stable object.

Each asteroid has the form of a convex polyhedron. To increase the chances of success of the experiment ACM wants to bring asteroids together in such manner that their centers of mass are as close as possible. To achieve this, ACM operators can rotate the asteroids and move them independently before bringing them together.

Help ACM to find out what minimal distance between centers of mass can be achieved.

For the purpose of calculating center of mass both asteroids are considered to have constant density.

Input

The input contains two descriptions of convex polyhedra.

The first line of each description contains integer number n — the number of vertices of the polyhedron ($4 \le n \le 60$). The following n lines contain three integer numbers x_i, y_i, z_i each — the coordinates of the polyhedron vertices ($-10^4 \le x_i, y_i, z_i \le 10^4$). It is guaranteed that the given points are vertices of a convex polyhedron, in particular no point belongs to the convex hull of other points. Each polyhedron is non-degenerate.

The two given polyhedra have no common points.

Output

Output one floating point number — the minimal distance between centers of mass of the asteroids that can be achieved. Your answer must be accurate up to 10^{-5}.

Sample Input

```
8
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1
5
0 0 5
1 0 6
-1 0 6
0 1 6
0 -1 6
```

Sample Output

```
0.75
```

Fibonacci

Little John studies numeral systems. After learning all about fixed-base systems, he became interested in more unusual cases. Among those cases he found a *Fibonacci system*, which represents all natural numbers in an unique way using only two digits: zero and one. But unlike usual binary scale of notation, in the Fibonacci system you are not allowed to place two 1s in adjacent positions.

One can prove that if you have number $N = \overline{a_n a_{n-1} \dots a_1}_F$ in Fibonacci system, its value is equal to $N = a_n \cdot F_n + a_{n-1} \cdot F_{n-1} + \dots + a_1 \cdot F_1$, where F_k is a usual Fibonacci sequence defined by $F_0 = F_1 = 1$, $F_i = F_{i-1} + F_{i-2}$.

For example, first few natural numbers have the following unique representations in Fibonacci system:

$$\begin{aligned} 1 &= 1_F \\ 2 &= 10_F \\ 3 &= 100_F \\ 4 &= 101_F \\ 5 &= 1000_F \\ 6 &= 1001_F \\ 7 &= 1010_F \end{aligned}$$

John wrote a very long string (consider it infinite) consisting of consecutive representations of natural numbers in Fibonacci system. For example, the first few digits of this string are 110100101100010011010...

He is very interested, how many times the digit 1 occurs in the N-th prefix of the string. Remember that the *N-th prefix* of the string is just a string consisting of its first N characters.

Write a program which determines how many times the digit 1 occurs in N-th prefix of John's string.

Input

The input contains a single integer N ($0 \le N \le 10^{15}$).

Output

Output a single integer — the number of 1s in N-th prefix of John's string.

Sample Input

```
21
```

Sample Output

```
10
```

Game

A group of contestants sits at the round table and plays the following game to relieve anxiety before the start of NEERC 2007. The game is played with a single token that is given to one person at the beginning of the game. This person passes the token to the adjacent person on the left-hand side or to the adjacent person on the right-hand side with a certain probability. A person who receives the token does the same with his own probability and so on. The game ends when each person has received the token at least once. The last person who has received the token wins.

The problem is to find the probability of winning for the given person. The probability of passing the token to the left or to the right is individual for each person and is known in advance before the beginning of the game.

Contestants are numbered from 1 to n so that the person number 2 sits to the right of 1, the person number 3 sits to the right of 2, and so on. The person number 1 sits to the right of n. The game starts with the person whose number is specified in the input and your task is to find the probability of winning for the person number n.

Input

The first line of the input contains two integer numbers n and k ($2 \le n \le 50, 1 \le k < n$). n denotes the total number of contestants, k denotes the number of the person who has the token at the beginning of the game.

The second line of the input contains $n-1$ numbers that denote the probabilities p_i ($0.01 \le p_i \le 0.99$) of passing the token to the right for the persons numbered from 1 to $n-1$. The probability of passing the token to the left for the person number i is $1 - p_i$. The probabilities are given with at most 2 digits after decimal point.

Output

Write to the output a single number that denotes the probability of winning for the person number n with a precision of at least 6 digits after decimal point.

Sample Input

```
7 3
0.5 0.5 0.5 0.5 0.5 0.5
```

Sample Output

```
0.1666666667
```

Sample Input

```
3 1
0.3 0.6
```

Sample Output

```
0.3000000000
```

Sample Input

```
24 12
0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.5
0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
```

Sample Output

```
0.9800000000
```

Grachten

Damn! I did not only oversleep (and today is *the* contest day!) but I also got stuck somewhere in Delft on my way from the hotel to the contest site. Everywhere around me are grachten, these city-canals that are part of many cities in the Netherlands. I am in a bit of hurry, because the NWERC contest starts in a few minutes.

To make matters even worse, some bridges in Delft are closed due to a cycling race through the city. Thus, I decided to jump over some of the grachten instead of searching for open bridges.

Everyone knows that computer scientists like me are good at algorithms but not very good athletes. Besides, I am a bit faint-hearted and don't want to get wet. So I need your help to calculate the distance I have to jump over a gracht.

Luckily, I did attend the excursion in Delft city center yesterday, where I learned that all paving stones in Delft are squares and have the same size. This way, I can do some measurements on my side of the gracht (my units are paving stones):

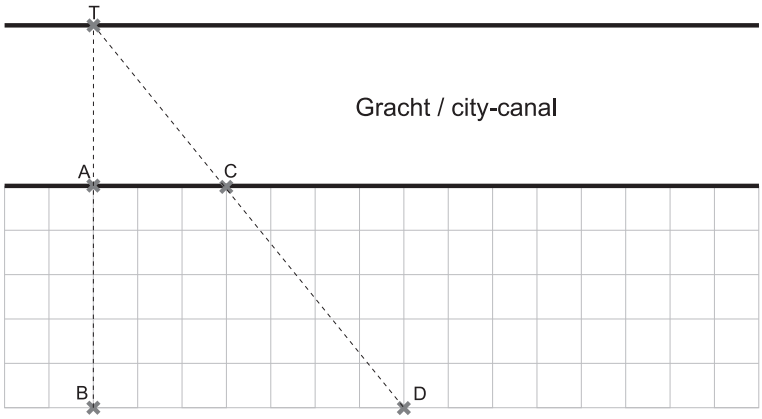


Figure 1: Illustration of first sample input.

I walked from point C to point D via points A and B while counting the paving stones.

Points A and C are always on the edge of the gracht. Points B and D have the same distance to the gracht. The target point T is always on the edge of the other side of the canal; it is the intersection point of the line through B and A, and the line through D and C. The angle between AT and AC is 90 degrees, and the two edges of the canal are parallel lines.

Please calculate the distance between A and T (necessary jump distance) for me.

Input

For each test case, the input consists of one line containing three positive integers that specify the distances between A and B, A and C, and B and D.

You may safely assume that no distance is larger than 1000 and the distance between B and D is larger than the distance between A and C.

Output

For each test case, print one line of output: the distance between A and T as a **reduced** fraction (i.e. remove all common factors of numerator and denominator).

Sample Input

```
5 3 7
5 3 8
1 2 3
23 42 47
500 500 1000
1 1 1000
```

Sample Output

```
15/4
3/1
2/1
966/5
500/1
1/999
```

Interconnect

There are two serious problems in the Kingdom of Lipshire: the roads and the fools who build them. Once upon a time, the King of Lipshire has decided to improve the road system because some roads became completely impassable – it was easier to travel cross-country instead of using those roads.

By Kings decree, new roads are to be built in Lipshire. Of course, the new road system must interconnect all towns, i. e. there must be a path connecting any two towns of Lipshire.

The road administration of Lipshire has resources to build exactly one road per year. Unfortunately, the fools who build these roads are completely out of control. So, regardless of the orders given, the fools randomly select two different towns a and b and build a road between them, even when those towns are already connected by a road. All possible choices are equiprobable. The road is build in such a manner that the only points where a traveler can leave it are the towns connected by this road. The only good thing is that all roads are bidirectional.

The King knows about the problem, but he cannot do anything about it. The only thing King needs to know is the expected number of years to wait before the road system of Lipshire becomes interconnected. He asked you to provide this information.

Input

The first line of the input contains two integers n and m ($2 \leq n \leq 30, 0 \leq m \leq 1000$) – the number of towns in Lipshire, and the number of roads which are still good. The following m lines describe roads, one per line. Each road is described with two endpoints two integer numbers u_i and v_i. There can be multiple roads between two towns, but the road from a town to itself is not allowed.

Output

Output the expected number of years to wait for the interconnected road system. If the system is already interconnected, output zero as an answer. Output the number with at least six precise digits after the decimal point.

Sample Input

```
2 1
1 2
```

Sample Output

```
0.0
```

Sample Input

```
4 2
1 2
3 4
```

Sample Output

```
1.5
```

iSharp

You are developing a new fashionable language that is not quite unlike C, C++, and Java. Since your language should become an object of art and fashion, you call it $i^{\#}$ (spelled i-sharp). This name combines all the modern naming trends and also hints at how smart you are.

Your language caters for a wide auditory of programmers and its type system includes arrays (denoted with "`[]`"), references (denoted with "`&`"), and pointers (denoted with "`*`"). Those type constructors can be freely combined in any order, so that a pointer to an array of references of references of integers (denoted with "`int&&[]*`") is a valid type.

Multiple variables in $i^{\#}$ can be declared on a single line with a very convenient syntax where common type of variables is given first, followed by a list of variables, each optionally followed by additional variable-specific type constructors. For example, the following line:

```
int& a*[]&, b, c*;
```

declares variables a, b, and c with types "`int&&[]*`", "`int&`", and "`int&*`" correspondingly. Note, that type constructors on the right-hand sides of variables in $i^{\#}$ bind to variable and their order is reversed when they are moved to the left-hand side next to type. Thus "`int*& a`" is equivalent to "`int a&*`".

However, you discover that coding style with multiple variable declarations per line is confusing and is outlawed in many corporate coding standards. You decide to get rid of it and refactor all existing $i^{\#}$ code to a single variable declaration per line and always specify type constructor next to the type it refers to (instead of the right-hand side of variable). Your task it to write such refactoring tool.

Input

The input contains a single line with a declaration of multiple variables in $i^{\#}$. The line starts with a type name, followed by zero, one, or more type constructors, followed by a space, followed by one or more variable descriptors separated by "`,`" (comma) and space, and terminated by "`;`" (semicolon). Each variable descriptor contains variable name, followed by zero, one, or more type constructors.

Type name and variable names are distinct and consist of lowercase and uppercase English letters from "`a`" to "`z`" or "`A`" to "`Z`". The line contains at most 120 characters and does not contain any extra spaces.

Output

Write to the output a line for each variable declared in the input. For each variable write its declaration on a single line in the same format as in the input, but with all type constructors next to its type. Separate type with all type constructors from a variable name by a single space. Do not write any extra spaces.

Sample Input

```
int& a*[]&, b, c*;
```

Sample Output

```
int&&[]* a;
int& b;
int&* c;
```

Sample Input

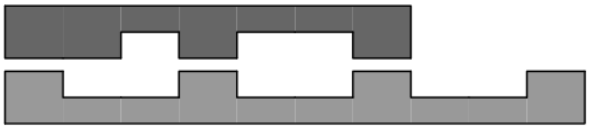
```
Double[][] Array[];
```

Sample Output

```
Double[][][] Array;
```

Kickdown

A research laboratory of a world-leading automobile company has received an order to create a special transmission mechanism, which allows for incredibly efficient kickdown – an operation of switching to lower gear. After several months of research engineers found that the most efficient solution requires special gears with teeth and cavities placed non-uniformly. They calculated the optimal flanks of the gears. Now they want to perform some experiments to prove their findings. The first phase of the experiment is done with planar toothed sections, not round-shaped gears. A section of length n consists of n units. The unit is either a cavity of height h or a tooth of height $2h$. Two sections are required for the experiment: one to emulate master gear (with teeth at the bottom) and one for the driven gear (with teeth at the top).



There is a long stripe of width $3h$ in the laboratory and its length is enough for cutting two engaged sections together. The sections are irregular but they may still be put together if shifted along each other.

The stripe is made of an expensive alloy, so the engineers want to use as little of it as possible. You need to find the minimal length of the stripe which is enough for cutting both sections simultaneously.

Input

There are two lines in the input, each contains a string to describe a section. The first line describes master section (teeth at the bottom) and the second line describes driven section (teeth at the top). Each character in a string represents one section unit – 1 for a cavity and 2 for a tooth. The sections can not be flipped or rotated. Each string is non-empty and its length does not exceed 100.

Output

Write a single integer number to the output – the minimal length of the stripe required to cut off given sections.

Sample Input

```
2112112112
2212112
```

Sample Output

```
10
```

Sample Input

```
12121212
21212121
```

Sample Output

```
8
```

Please, go first

You are currently on a skiing trip with a group of friends. In general, it is going well: you enjoy the skiing during the day and, of course, the apres-skiing during the night. However, there is one nuisance: the skiing lift. As always, it is too small, and can only serve one person every 5 seconds. To make matters worse, you and your friends generally dont arrive simultaneously at the lift, which means that you spend time waiting at the bottom of the mountain for the lift and at the top again for your friends.

The waiting at the top is especially inefficient. In fact, you realize that if your friends havent arrived yet, you might as well let other people pass you in the queue. For you, it makes no difference, since otherwise youd be waiting at the top. On the other hand, your actions might save them time if their friends have already arrived and are currently waiting for them at the top.

You are wondering how much time would be saved if everybody adopts this nice attitude. You have carefully observed the queue for a while and noticed which persons form groups of friends. Suppose someone lets another pass if doing this doesnt change his own total waiting time, but saves time for the other person. Do this over and over again until it cant be done anymore. How much time will this save, in total?

Input

On the first line a positive integer: the number of test cases, at most 100. After that per test case:

- one line with an integer n ($1 \leq n \leq 25000$): the number of people in the line for the lift.
- one line with n alphanumeric characters (uppercase and lowercase letters and numbers): the queue. The first person in this line corresponds to the person at the head of the queue. Equal characters correspond to persons from the same group of friends.

Output

Per test case: one line with an integer: the time saved, in seconds.

Sample Input

```
2
6
AABABB
10
Ab9AAb2bC2
```

Sample Output

```
15
45
```

Stealing Harry Potter's Precious

Harry Potter has some precious. For example, his invisible robe, his wand and his owl. When Hogwarts school is in holiday, Harry Potter has to go back to uncle Vernon's home. But he can't bring his precious with him. As you know, uncle Vernon never allows such magic things in his house. So Harry has to deposit his precious in the Gringotts Wizarding Bank which is owned by some goblins. The bank can be considered as a $N \times M$ grid consisting of $N \times M$ rooms. Each room has a coordinate. The coordinates of the upper-left room is (1,1), the bottom-right room is (N,M) and the room below the upper-left room is (2,1).

Some rooms are indestructible and some rooms are vulnerable. Goblins always care more about their own safety than their customers' properties, so they live in the indestructible rooms and put customers' properties in vulnerable rooms. Harry Potter's precious are also put in some vulnerable rooms. Dudely wants to steal Harry's things this holiday. He gets the most advanced drilling machine from his father, uncle Vernon, and drills into the bank. But he can only pass though the vulnerable rooms. He can't access the indestructible rooms. He starts from a certain vulnerable room, and then moves in four directions: north, east, south and west. Dudely knows where Harry's precious are. He wants to collect all Harry's precious by as less steps as possible. Moving from one room to another adjacent room is called a 'step'. Dudely doesn't want to get out of the bank before he collects all Harry's things. Dudely is stupid. He pay you $1,000,000 to figure out at least how many steps he must take to get all Harry's precious.

Input

There are several test cases. In each test cases: The first line are two integers N and M, meaning that the bank is a $N \times M$ grid ($0 < N, M \leq 100$). Then a $N \times M$ matrix follows. Each element is a letter standing for a room. '#' means a indestructible room, '.' means a vulnerable room, and the only '@' means the vulnerable room from which Dudely starts to move. The next line is an integer K ($0 < K \leq 4$), indicating there are K Harry Potter's precious in the bank. In next K lines, each line describes the position of a Harry Potter's precious by two integers X and Y, meaning that there is a precious in room (X,Y). The input ends with N = 0 and M = 0.

Output

For each test case, print the minimum number of steps Dudely must take. If Dudely can't get all Harry's things, print -1.

Sample Input

```
2 3
##@
#.#
1
2 2
4 4
#@##
....
####
....
2
2 1
2 4
0 0
```

Sample Output

```
-1
5
```

Wally World

Two star-crossed lovers want to meet. The two lovers are standing at distinct points in the plane (but then again, aren't we all?). They can travel freely except that there is a single wall which cannot be crossed. The wall is a line segment which is parallel to either the x or y axis. Each lover can move 1 unit in 1 second. How long will it take them to be together if they both choose the best path?

Input

Input for each test case will consist of two lines each containing four integers. The first two integers will specify the x and y coordinates of the first lover; the next two integers will specify the x and y coordinates of the second lover. The next four integers will specify the start and end points of the wall. Furthermore, in all cases both lovers will not be on the (infinite) line containing the wall — that is, the wall extended in both directions. All coordinates will be positive and less than or equal to 10000 and neither lover will start on the wall. The input will be terminated by a line containing four zeroes.

Output

For each test case, output the minimum time in seconds for the two lovers to meet. Print the answer to exactly 3 decimal places, using the output format shown in the example.

Sample Input

```
5 2 7 2
1 1 1 100
1 2 3 2
2 1 2 100
0 0 0 0
```

Sample Output

```
Case 1: 1.000
Case 2: 1.414
```

Yuangfang, what do you think?

Factoring a polynomial is always a hard and important issue in mathematics teaching in middle schools. Teacher Liu loves teaching this issue very much, but his students are not good at it. Yuanfang is the best student in Teacher Liu's class. Every time when Teacher Liu comes up with a hard problem and it seems no student can solve it, Liu always says: "Yuanfang, what do you think?". This week, Teacher Liu began to teach how to factor a polynomial.
On Monday, Teacher Liu said: "Let's factor $x^2 - 1$ ... Yuangfang, what do you think?"
On Tuesday, Teacher Liu said: "Let's factor $x^3 - 1$ ... Yuangfang, what do you think?"
On Wednesday, Teacher Liu said: "Let's factor $x^4 - 1$ ... Yuangfang, what do you think?" ...

On Friday, Yuanfang got crazy. She wanted to solve this problem permanently. So she came to you, the only programmer she knows, for help. You should write a program to factor the polynomial $x^n - 1$. In other words, represent the polynomial $x^n - 1$ by a product of irreducible polynomials in which coefficients are all integers.

Input

There are several test cases. Every case is an integer n in a line ($n \leq 1100$) meaning that you should factor the polynomial $x^n - 1$.

Output

We print polynomials like this:
1. x^2 to x^2;
2. $x^3 - 1$ to x^3-1;
3. $x^6 - 2x^4 + 1$ to x^6-2x^4+1.

For each test case, you should print the result polynomials in a certain order. To sort the polynomials, we compare the coeffcients of them from high-degree to low-degree. The coeffcient with a smaller absolute value has a smaller order. When absolute values are the same, negative coeffcient has a smaller order. Please print the result polynomials from small order to large order. In the result, put every factor polynomial between a pair of parentheses (except that the result is just $x - 1$ as shown in sample.

Every factor polynomial must satisfy the following conditions:
1. It can't be factored any more.
2. All the terms of the same degree must be combined.
3. No term's coeffcient is 0.
4. The terms appear in the descending order by degree.
5. All coeffcients are integers.

Sample Input

```
1
2
3
4
5
6
0
```

Sample Output

```
x-1
(x-1)(x+1)
(x-1)(x^2+x+1)
(x-1)(x+1)(x^2+1)
(x-1)(x^4+x^3+x^2+x+1)
(x-1)(x+1)(x^2-x+1)(x^2+x+1)
```