Problem G
Research Productivity Index

Angela is a new PhD student and she is nervous about the upcoming paper submission deadline of this year's research conference. She has been working on multiple projects throughout the past year. Luckily most of the projects concluded successfully, and she came up with n candidate papers. However not all of the papers were born equal—some have better results than others. Her advisor believes she should only submit the papers with "good enough" results so they have a high chance of getting accepted.

Angela's research group has a unique way of evaluating the success of paper submissions. They use the *research productivity index*, defined as $a^{a/s}$, where s is the total number of papers submitted, and a is the number of papers that are accepted by the conference. When $a = 0$, the index is defined to be zero. For example:

- if one paper is submitted and it gets accepted, the index is $1^{1/1} = 1$;
- if 4 papers are submitted and all get accepted, the index is $4^{4/4} = 4$;
- if 10 papers are submitted and 3 get accepted, the index is $3^{3/10} \approx 1.390389$;
- if 5 papers are submitted and 4 get accepted, the index is $4^{4/5} \approx 3.031433$;
- if 3 papers are submitted and all get rejected ($a = 0$), the index is 0.

Intuitively, to get a high research productivity index one wants to get as many papers accepted as possible while keeping the acceptance rate high.

For each of her n papers, Angela knows exactly how likely it is that the conference would accept the paper. If she chooses wisely which papers to submit, what is the maximum expected value of her research productivity index?

Input

The first line of the input has a single integer n ($1 \leq n \leq 100$), the number of Angela's candidate papers. The next line has n space-separated integers giving the probability of each paper getting accepted. Each probability value is given as an integer percentage between 1 and 100, inclusive.

Output

Output the maximum expected value of Angela's research productivity index. Your answer is considered correct if it has an absolute or relative error of no more than 10^{-6}.

Sample Input 1	Sample Output 1
5 30 50 70 60 90	2.220889579

Sample Input 2	Sample Output 2
6 30 90 30 90 30 90	2.599738456

Sample Input 3	Sample Output 3
4 10 10 10 10	0.368937005

Problem H
Running Routes

The administrators at Polygonal School want to increase enrollment, but they are unsure if their gym can support having more students. Unlike a normal, boring, rectangular gym, the gym floor at Polygonal is a regular n-sided polygon! They affectionately refer to the polygon as P.

The coach has drawn several running paths on the floor of the gym. Each running path is a straight line segment connecting two distinct vertices of P. During gym class, the coach assigns each student a different running path, and the student then runs back and forth along their assigned path throughout the class period. The coach does not want students to collide, so each student's path must not intersect any other student's path. Two paths intersect if they share a common point (including an endpoint).

Given a description of the running paths in P, compute the maximum number of students that can run in gym class simultaneously.

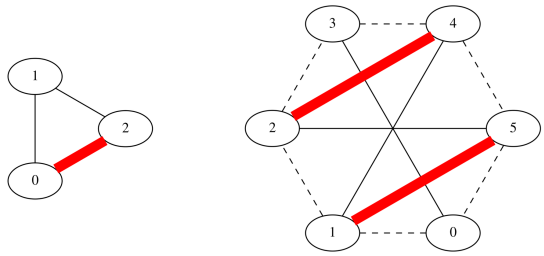

Figure H.1: Illustrations of the two sample inputs, with possible solutions highlighted in thick red lines. Solid black lines represent running paths that are not assigned to a student, and dashed black lines are used to show the boundary of P in places where no running path exists.

Input

The first line contains an integer n ($3 \leq n \leq 500$), the number of vertices in P. (The vertices are numbered in increasing order around P.) Then follows n lines of n integers each, representing a $n \times n$ symmetric binary matrix which we'll call M. The j^{th} integer on the i^{th} line M_{ij} is 1 if a running path exists between vertices i and j of the polygon, and 0 otherwise. It is guaranteed that for all $1 \leq i, j \leq n$, $M_{ij} = M_{ji}$ and $M_{ii} = 0$.

Output

Print the maximum number of students that can be assigned to run simultaneously on the running paths, given the above constraints.

Sample Input 1

```
3
0 1 1
1 0 1
1 1 0
```

Sample Output 1

```
1
```

Sample Input 2

```
6
0 0 0 1 0 0
0 0 0 0 1 1
0 0 0 0 1 1
1 0 0 0 0 0
0 1 1 0 0 0
0 1 1 0 0 0
```

Sample Output 2

```
2
```

Problem I
Slow Leak


TheDigitalWay

You are an avid cyclist and bike every day between your home and school. Usually, your ride is uneventful and you bike along the shortest path between home and school. After school this afternoon you realized you have a slow leak in your bike tire—the tire can hold some air, but not for long. Refilling the tire allows you to ride your bike for some distance, after which your tire goes flat again and it becomes impossible to ride any further (and you refuse to walk your bicycle).

Luckily for you, your city has installed several bike repair stations at intersections throughout town where you can refill your tire and bike again until the tire goes flat. There's a repair station at your school too, so that you can fill up your tire before you start on your trip home.

You've calculated how far you can bike before your tire runs out of air and you know the layout of your town, including all the intersections, distances between them, and the locations of the repair stations. What is the shortest possible trip from school to your home that you can take without becoming stuck due to a flat tire? (You do not become stuck if you roll into a repair station, or your home, at the exact same time as your tire goes flat.)

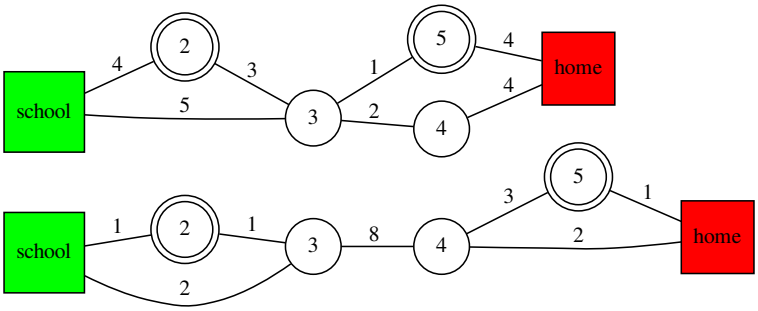

Figure I.1: An illustration of the two sample inputs.

Input

The first line of input contains four integers n, m, t, and d, satisfying $2 \leq n \leq 500$, $0 < m \leq n(n-1)/2$, $0 < t < n$ and $0 < d < 2^{31}$. The value n represents the number of intersections in the city, m represents the number of roads in the city, t is the number of repair stations and d is the distance that you can bike (starting with a fully inflated tire) before your tire goes flat again. The intersections are numbered from 1 to n. Your school is at intersection 1 and your home is at intersection n.

The second line of input contains t integers, with values between 2 and $n-1$, inclusive. These are the intersections

where the repair stations are located (excluding your school's repair station). All integers on this line are unique.

The next m lines represent the roads in your town. Each has three integers i, j, k, where $1 \leq i < j \leq n$, and $0 < k < 2^{31}$. These three integers represent that there is a direct road from intersection i to intersection j of length k. Roads can be traveled in either direction. There is at most one road between any two intersections.

Output

Print the minimum total distance you need to travel to reach home from school without getting stuck due to a flat tire. If the trip is not possible, output the word `stuck` on a single line instead.

It is guaranteed that if the trip is possible, the minimum distance D satisfies $0 < D < 2^{31}$.

Sample Explanation

In the first sample input, if your tire did not have a leak then the shortest path home would have a distance of 9, going from the school through intersections 3 and 5. However, due to the leak, you can only travel a distance of 4 before you need to refill the tire, requiring you to use the repair stations at intersections 2 and 5, for a total distance of 12.

In the second sample input, if your tire did not have a leak, then the shortest path home would have a distance of 12. But since your tire only lasts for a distance of 10, there's no path where your bike tire will not go flat somewhere along the way. Even when using repair station at intersection 2, you get stuck before you can reach either your home or the repair station at intersection 5.

Sample Input 1	Sample Output 1
6 7 2 4 2 5 1 2 4 1 3 5 2 3 3 3 4 2 3 5 1 4 6 4 5 6 4	12

Sample Input 2	Sample Output 2
6 7 2 10 2 5 1 2 1 1 3 2 2 3 1 3 4 8 4 5 3 4 6 2 5 6 1	stuck

Problem J
Stop Counting!

Photo by Drew Rae

The Martingale casino is creating new games to lure in new gamblers who tire of the standard fare. Their latest invention is a fast-paced game of chance called *Stop Counting!*, where a single customer plays with a dealer who has a deck of cards. Each card has some integer value.

One by one, the dealer reveals the cards in the deck in order, and keeps track of the sum of the played cards and the number of cards shown. At some point before a card is dealt, the player can call "Stop Counting!" After this, the dealer continues displaying cards in order, but does not include them in the running sums. At some point after calling "Stop Counting!", and just before another card is dealt, the player can also call "Start Counting!" and the dealer then includes subsequent cards in the totals. The player can only call "Stop Counting!" and "Start Counting!" at most once each, and they must call "Stop Counting!" before they can call "Start Counting!". A card is "counted" if it is dealt before the player calls "Stop Counting!" or is dealt after the player calls "Start Counting!"

The payout of the game is then the average value of the counted cards. That is, it is the sum of the counted cards divided by the number of counted cards. If there are no counted cards, the payout is 0.

You have an 'in' with the dealer, and you know the full deck in order ahead of time. What is the maximum payout you can achieve?

Input

The first line of the input contains a single integer $1 \leq N \leq 1\,000\,000$, the number of cards in the deck.

The second line of input contains N space-separated integers, the values on the cards. The value of each card is in the range $[-10^9, 10^9]$. The cards are dealt in the same order they are given in the input.

Output

Output the largest attainable payout. The answer is considered correct if the absolute error is less than 10^{-6}, or the relative error is less than 10^{-9}.

Sample Explanation

In the first sample, by calling "Stop Counting!" before the -10 and "Start Counting!" before the final 10, we can achieve an average of 10.0 with the cards that are counted.

In the second sample, all values are negative, so the best strategy is to call "Stop Counting!" before the first card is dealt and call "Start Counting!" after the last card is dealt. Since no cards were counted, the average of the counted cards is 0.0.

Sample Input 1	Sample Output 1
5 10 10 -10 -4 10	10.000000000

Sample Input 2	Sample Output 2
4 -3 -1 -4 -1	0.000000000

Problem K
Summer Trip

Cartoon of items related to travel. Image from mohamed_hassan at pixabay.com

Leo has started a job in a travel agency. His first task is to organize a summer trip to an exotic overseas city. During the summer season, events of various types take place in the city: sports matches, concerts, beach parties, and many others. At any given time, there is exactly one event taking place. Events of any particular type may take place more than once during the season. The itinerary of events that Leo offers to his clients cannot be chosen arbitrarily; the company requires them to form a so-called "good itinerary." A good itinerary is a consecutive sequence of at least two events in the summer season, where the first and last events are of different types, and they are both unique among all event types during the sequence. For example, if the first event in a good itinerary is a beach party, none of the other events during the itinerary can also be a beach party. There are no other restrictions on the event types in the sequence of a good itinerary.

Before he starts organizing the trip, Leo wants to know the total number of good itineraries that are possible given a calendar of events that will take place over the summer season.

Input

The input consists of one line with a string describing the sequence of event types in the summer season. All characters are lowercase English letters (`'a'–'z'`), with different letters represent different types of events. Character i of the string encodes the ith event of the summer. There are no blanks or spaces in the string.

The length of the input string is at least 2 and at most 100 000 characters.

Output

Print the number of good itineraries that exist for the given summer season.

Sample Input 1	Sample Output 1
abbcccddddeeeee	10

Sample Input 2	Sample Output 2
thenumberofgoodstringsis	143

This page is intentionally left blank.

Problem L
Traveling Merchant

Lithograph by Louis Haghe from an original by David Roberts, Wikipedia

There is a long east-west road which has n towns situated along it, numbered 1 to n from west to east. All towns buy and sell the same kind of goodie. The value of a goodie fluctuates according to a weekly schedule. A town buys and sells a goodie at its value in that town on that particular day. At town i, the value of a goodie changes by d_i every day in the first half of a week, and changes by $-d_i$ every day in the second half of a week. In other words, the value of a goodie at town i is v_i on Mondays and Sundays, $v_i + d_i$ on Tuesdays and Saturdays, $v_i + 2d_i$ on Wednesdays and Fridays, and $v_i + 3d_i$ on Thursdays.

A merchant is making a business travel plan. His trip begins at a starting town s and ends at a destination town t, visiting each town from s to t (inclusive) exactly once. The merchant starts the trip on a Monday. It takes him exactly one day to travel between adjacent towns and every day he travels to the next town on the path to the destination. He may buy exactly one goodie at a town along the trip and sell that goodie at a town he visits later. He can only buy once and sell once. The merchant would like to know the maximum possible profit of q travel plans with different choices of town s and town t.

Input

The first line of the input has a single integer n ($2 \le n \le 10^5$). The next n lines each have two integers. The ith line has v_i ($1 \le v_i \le 10^9$) and d_i ($1 \le v_i + 3d_i \le 10^9$). The next line has a single integer q ($1 \le q \le 10^5$). The following q lines each give a pair of integers s and t ($1 \le s, t \le n, s \ne t$), representing one travel plan from town s to town t. If $s < t$ the merchant travels west to east, otherwise he travels east to west.

Output

For each travel plan, output the maximum profit the merchant can make on a single line. If the merchant cannot make any profit, output 0.

Sample Input 1	Sample Output 1
5 1 2 2 1 5 0 4 -1 7 -2 5 1 5 5 1 3 1 4 5 5 4	4 2 2 1 0

Problem M
Zipline

Photo by Virginia State Parks.

A zipline is a very fun and fast method of travel. It uses a very strong steel cable, connected to two poles. A rider (which could be a person or some cargo) attaches to a pulley which travels on the cable. Starting from a high point on the cable, gravity pulls the rider along the cable.

Your friend has started a company which designs and installs ziplines, both for fun and for utility. However, there's one key problem: determining how long the cable should be between the two connection points. The cable must be long enough to reach between the two poles, but short enough that the rider is guaranteed to stay a safe distance above the ground. Help your friend determine these bounds on the length.

The cable connects to two vertical poles that are w meters apart, at heights g and h meters, respectively. You may assume that the cable is inelastic and has negligible weight compared to the rider, so that there is no sag or slack in the cable. That is, at all times the cable forms two straight line segments connecting the rider to the two poles, with the sum of the segments lengths equal to the total length of the cable. The lowest part of the rider hangs r meters below the cable; therefore the cable must stay at least r meters above the ground at all times during the ride. The ground is flat between the two poles. Please refer to the diagram in Figure M.1 for more information.

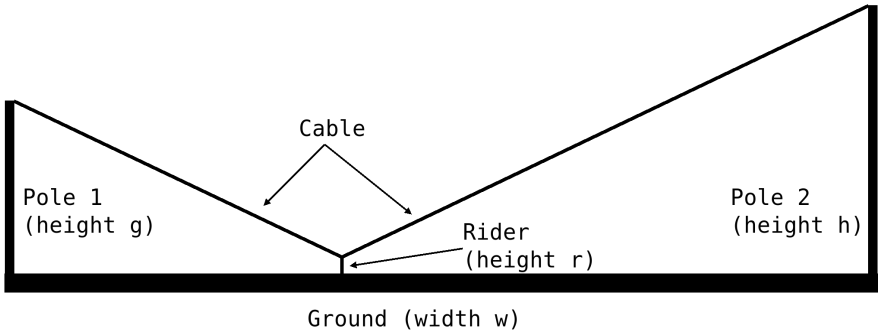

Figure M.1: A zipline, annotated with the four variables used to describe it.

Input

The input starts with a line containing an integer n, where $1 \leq n \leq 1\,000$. The next n lines each describe a zipline configuration with four integers: w, g, h, and r. These correspond to the variables described above. The limits on their values are: $1 \leq w, g, h \leq 1\,000\,000$, and $1 \leq r \leq \min(g, h)$.

Output

For each zipline, print a line of output with two lengths (in meters): the minimum and maximum length the cable can be while obeying the above constraints. Both lengths should have an absolute error of at most 10^{-6}.

Sample Input 1	Sample Output 1
2 1000 100 100 20 100 20 30 2	1000.00000000 1012.71911209 100.49875621 110.07270325