ICPC SOUTH PACIFIC REGIONALS

DECEMBER 11, 2021

Contest Problems

A: Antiwork
B: Brand New PC
C: Cardinal Lanterns
D: Dog Catcher
E: Explosive Wiring 2
F: Fantasy Hockey Trades
G: Gungle 2
H: Handy Printer
I : Incredibly Average
J : Jack and Jill Jotting Notes
K: Keyboard
L: Laser Turrets

This contest contains twelve problems over 28 pages. Good luck.

For problems that state "*Your answer should have an absolute or relative error of less than* 10^{-n}", your answer, x, will be compared to the correct answer, y. If $|x - y| < 10^{-n}$ or $\frac{|x-y|}{|y|} < 10^{-n}$, then your answer will be considered correct.

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \leq a < m$
  and
- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \leq \min(n, \ell)$ such that $s_i = t_i$ for all $1 \leq i < k$ and $s_k < t_k$
  or
- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters $(A, B, \ldots, Z)$.
- Lowercase letters are the lowercase English letters $(a, b, \ldots, z)$.

Definition 4

Unless otherwise specified, the distance between two points (x_0, y_0) and (x_1, y_1) is defined as its Euclidean distance:

$$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}.$$

Problem A
Antiwork
Time limit: 2 seconds

Have you heard the big news? Harry has decided to work exactly K more days until he quits his job.

Harry lives in a city with N intersections connected by M directed roads. Harry's house is located at the intersection with label 1, and his work is at the intersection with label N.

It is boring to take the same route to work. So, Harry wants to walk a different route to work each of the K days. Because it is inefficient to visit an intersection or road multiple times, each route must not contain the same road or intersection more than once. It is okay to visit the same intersection or road on two different days, just not on the same day. Harry teleports home from work, so there is no need to find two routes per day.

Input

The first line of input contains three integers, N $(1 \leq N \leq 50)$, M $(0 \leq M \leq N^2 - N)$, and K $(1 \leq K \leq 50)$. The next M lines describe the roads. Each line contains two distinct integers, u $(1 \leq u \leq N)$ and v $(1 \leq v \leq N)$ which indicates that there is a directed road from intersection u to v. No two roads connect the same intersections in the same direction.

Output

If it is not possible to find K distinct routes to work, display `Stay at home`. Otherwise, display K distinct routes to work, each on a separate line. Each route to work should start with the number of intersections in the route, followed by the intersection labels comprising the route in order.

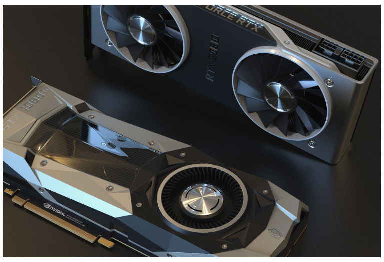
Sample Input 1	Sample Output 1
2 1 1 1 2	2 1 2

Sample Input 2	Sample Output 2
3 3 2 1 2 2 3 3 1	Stay at home

Sample Input 3	Sample Output 3
4 4 2 1 2 2 4 1 3 3 4	3 1 2 4 3 1 3 4

This page is intentionally left (almost) blank.

Problem B
Brand New PC
Time limit: 8 seconds



Bridget is trying to name her new gaming PC (somehow she was able to obtain a new GPU in 2021). She has not been able to make up her mind about the name. She started with a name S containing N lowercase letters. Then she changed her mind M times.

Each time Bridget changed her mind, she modified the previous name by changing the letter at a particular index. The changes were cumulative. For example, she may have started with a name "`alice`", then modified the second letter to make "`aoice`", then the first to make "`boice`", then the third to make "`bobce`". The result of this is that Bridget considered $M + 1$ different names including the name she started with.

Since keeping track of all the names she has come up with is getting difficult, Bridget has decided to order them all lexicographically.

Input

The first line of input contains two integers, N ($1 \leq N \leq 200\,000$), which is the length of the string, and M ($1 \leq M \leq 200\,000$), which is the number of modifications. The next line of input contains a single string, the name S. The string contains only lowercase letters and is of length N. The remaining M lines of input each contain an integer i ($1 \leq i \leq N$) followed by a lowercase letter l. The integer i is an index into the current name and indicates that the letter at that index was replaced by l.

Output

Display $M + 1$ integers representing the lexicographically sorted names. The first name is 0, the name after the first modification is 1, and so on. Ties should be broken by having the earliest name (in modification order) come first.

Sample Input 1	Sample Output 1
2 2 ab 1 b 1 a	0 2 1

Sample Input 2	Sample Output 2
4 3 icpc 4 h 3 i 2 a	3 2 0 1

Sample Input 3	Sample Output 3
3 3 aba 2 a 2 b 2 a	1 3 0 2

This page is intentionally left (almost) blank.

Problem C
Cardinal Lanterns
Time limit: 4 seconds

Glenda is the governor of Gridtown, and she loves her new lanterns: The Luminator 9000. Gridtown is made up of R roads that run east-west and C roads that run north-south creating exactly $R \cdot C$ intersections. On each intersection, Glenda may or may not place a lantern.

If a lantern is placed on an intersection, it will illuminate that intersection as well as all other intersections that are on the same road. This includes all intersections on both the east-west horizontal road and north-south vertical road containing the lantern.

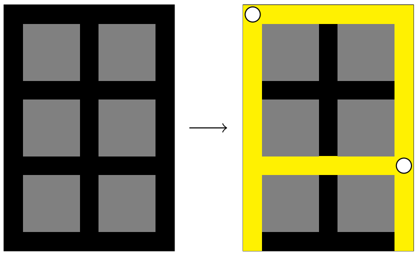


Figure C.1: The first sample.

Given the location of all lanterns, help Glenda by counting the number of illuminated intersections.

Input

The first line of input contains three integers N ($1 \leq N \leq 200\,000$), which is the number of lanterns, R ($1 \leq R \leq 10^{18}$), which is the number of roads running east-west, and C ($1 \leq C \leq 10^{18}$ and $1 \leq R \cdot C \leq 10^{18}$), which is the number of roads running north-south.

The next N lines describe the lanterns. Each of these lines contains two integers r ($1 \leq r \leq R$), which is the index of the east-west road this lantern is on, and c ($1 \leq c \leq C$), which is the index of the north-south road this lantern is on.

Output

Display the number of illuminated intersections.

Sample Input 1	Sample Output 1
2 4 3 1 1 3 3	10

Sample Input 2	Sample Output 2
3 3 5 1 2 2 4 3 3	15

Sample Input 3	Sample Output 3
1 3 2 2 2	4

This page is intentionally left (almost) blank.

Problem D
Dog Catcher
Time limit: 5 seconds

Dani the Dog Walker is at the dog park watching the dogs run around and play. It's almost the end of the day, so Dani must catch all the dogs so she can bring them back to their loving homes.

The dog park has N play areas that are connected by exactly $N - 1$ bi-directional pathways. There is a way to walk from every play area to every other play area using the pathways. Dani is currently standing at a play area that has exactly one pathway directly connected to it, and the dogs are at some other play areas in the park. It is fine if multiple dogs are at the same play area as each other.

Dani and her dogs treat this as a game. She and the dogs alternate turns with the dogs going first.

On the dogs' turn, each dog can either stay put or run to any play area in the dog park they can reach via the pathways as long as they do not go through the play area Dani is currently at.

On Dani's turn, she can move to any play area that is directly connected to the place she is now, by moving through one pathway. If there are dogs there, she catches all of them.

All the dogs start in a different play area to Dani. Dani wants to minimize the total time to catch all the dogs. The dogs want to play, and therefore want to maximize the total time Dani takes to catch all the dogs.

It can be shown that no matter how the dogs move, Dani can always eventually catch all of the dogs. If the dogs (as a group) and Dani play optimally, how many turns will it take Dani to catch all of the dogs?

Input

The first line of input contains two integers N ($2 \leq N \leq 50$), which is the number of play areas, and D ($1 \leq D \leq 50$), which is the number of dogs.

The next $N-1$ lines describe the pathways. Each of these lines contains two integers a and b ($1 \leq a \leq b \leq N$), which means that there is a pathway directly connecting play areas a and b.

Dani starts at play area 1. There is exactly 1 pathway connected to play area 1.

Output

Display the number of turns it will take Dani to catch all of the dogs.

Sample Input 1	Sample Output 1
2 5 1 2	1

Sample Input 2	Sample Output 2
4 10 1 2 2 3 2 4	8

This page is intentionally left (almost) blank.

Problem E
Explosive Wiring 2
Time limit: 5 seconds

If you were at the 2019 South Pacific ICPC Regional Finals, you might have helped design the wiring for a new kind of computer chip. You are back at it again and you have designed an even better wiring system for an even better computer chip. This time, your chip supports having a lot more wires than before but has them organized in a simpler way than the old design. Unfortunately, the wires are still made out of the same strange material that will explode under the wrong conditions.

You have a set of wires that you can install on the chip. All the wires are parallel to each other and each wire connects two points on the chip. A wire can be represented as an interval between A_i and B_i. This means that the wire starts at the point A_i micrometers from the left edge of the chip and ends B_i micrometers from the left edge of the chip. All of the endpoints (start or end locations) of the wires are unique.

Two wires i and j *interfere* if their intervals overlap (that is there is at least one value X such that $A_i \leq X \leq B_i$ and $A_j \leq X \leq B_j$). A wire does not interfere with itself. A set of wires is *safe* if each wire in the set interferes with **exactly one** other wire in the set. Given a set of wires from which to choose, what is the largest number of wires you can choose such that the set of chosen wires is safe?

Input

The first line of input contains a single integer N ($2 \leq N \leq 300\,000$), which is the number of wires.

The next N lines describe the wires. Each of these lines contains two integers A_i and B_i ($1 \leq A_i < B_i \leq 10^6$), which represents that this wire spans the interval from A_i to B_i on the chip.

All endpoints in the input are distinct.

Output

Display the largest number of wires in a safe set.

Sample Input 1	Sample Output 1
4 1 5 3 6 4 10 8 9	2

Sample Input 2	Sample Output 2
4 1 5 3 6 7 10 8 9	4

This page is intentionally left (almost) blank.

Problem F
Fantasy Hockey Trades
Time limit: 1 second

One of the true pleasures in life is participating in fantasy hockey! Each *manager* in the fantasy hockey league contracts with several athletes. Each athlete has a contract with exactly one manager. As a manager yourself, you are looking to make the best team by contracting the players you project will perform the best.

Managers in the league are allowed to trade athletes with each other. For example, say Felicity originally contracts the players $\{A, B, C, D\}$, and you originally contract the players $\{W, X, Y, Z\}$. You and Felicity can decide to trade players $\{W, X\}$ for players $\{A, C\}$. After the trade, Felicity would manage players $\{B, D, W, X\}$, while you would manage players $\{A, C, Y, Z\}$. A trade must always send the same number of players in each direction.

Every manager in the league has a projection of the number of points they think each athlete will get over the next year. For example, Felicity may project that Player X will receive 100 points, while you may project that Player X will receive 120 points.

You have figured out all of Felicity's projections and you know that they will accept any trade which makes their team better, according to their projections. After such a trade, Felicity will project to score strictly more points. Such a trade is called *fair*.

Consider the following case, where you and Felicity each contract three players:

Your team			Felicity's team		
	Your Projection	Felicity's Projection		Your Projection	Felicity's Projection
Sakic	10	10	Howe	10	20
Bourque	15	25	Orr	20	20
Lemieux	30	25	Gretzky	35	25

In this case, you and Felicity should trade Bourque and Lemieux for Orr and Gretzky. According to your projections, your team now has 65 points, and according to Felicity's projections, their team now has 70 points. Since both teams increased in projected value (yours from 55 to 65 and Felicity's from 65 to 70), this trade is fair.

Your goal is to maximize your team's points using your projections. You do not care how good you make Felicity's team. Knowing both your projections and Felicity's projections, what is the maximum number of points your team can achieve after at most one fair trade?

Input

The first line of input contains a single integer N ($1 \leq N \leq 10$), which is the number of athletes on each team.

The next N lines describe the athletes on your team. Each of these lines contains two integers a ($1 \leq a \leq 100\,000\,000$), which is your projection of the number of points for this player, and b ($1 \leq b \leq 100\,000\,000$), which is Felicity's projection of the number of points for this player.

The next N lines describe the athletes on Felicity's team. Each of these lines contains two integers a ($1 \leq a \leq 100\,000\,000$), which is your projection of the number of points for this player, and b ($1 \leq b \leq 100\,000\,000$), which is Felicity's projection of the number of points for this player.

Output

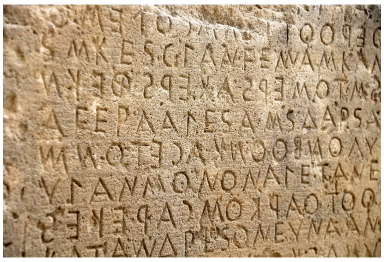
Display the maximum number of points you can project for your team after at most one fair trade.

Sample Input 1	Sample Output 1
3 10 10 15 25 30 25 10 20 20 20 35 25	65

Sample Input 2	Sample Output 2
1 10 15 20 12	20

Sample Input 3	Sample Output 3
1 10 1 2 20	10

Problem G
Gungle Expert
Time limit: 1 second



Since the South Pacific ICPC Divisional Finals of 2021, Gungle has become a sensation. You are training to become a Gungle expert.

In the game of Gungle, you must construct a word consisting of only uppercase letters. If you are able to make a word in which each vowel comes immediately after either a 'G' or an 'L', then that word is called a *Gungle* and you win. The vowels are A, E, I, O, and U.

You have a word W. You must change the minimum number of characters in W so that you have a Gungle. You may not add or remove characters, only change them. Given W, find a word that has the minimum number of characters changed that is a Gungle.

Input

The input consists of a single line containing a single string W ($1 \leq |W| \leq 100$). W only contains uppercase letters.

Output

Display a string that has the minimum number of changes from W and is a Gungle. If there are multiple correct solutions, any will be accepted.

Sample Input 1	Sample Output 1
GUNGLE	GUNGLE

Sample Input 2	Sample Output 2
AAAA	GAGA

Sample Input 3	Sample Output 3
ICPC	GCPC

This page is intentionally left (almost) blank.

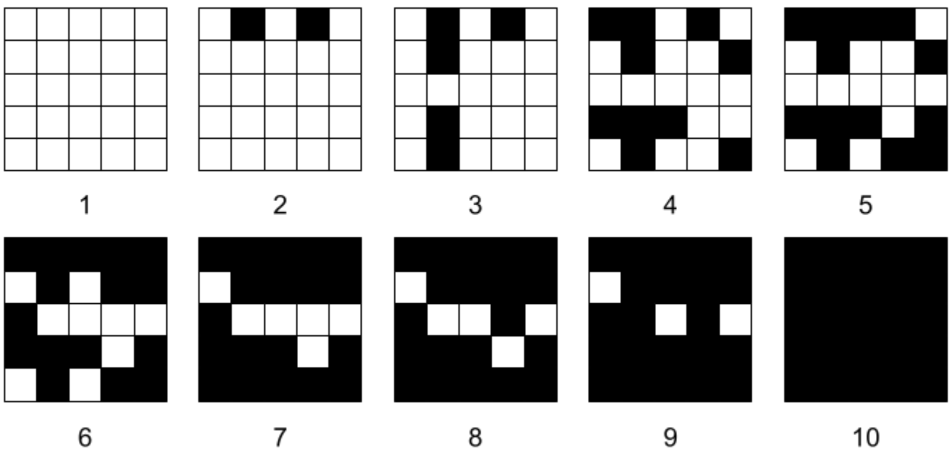
Problem H
Handy Printer
Time limit: 3 seconds

Rie has a special printer that operates in an unusual way. It takes an instruction matrix, which is a 5×5 matrix of integers in the range 0 to 1000, inclusive. It then prints 1000 black and white images that are each 5×5. The images are numbered 1 to 1000. In the i-th image, if an element of the instruction matrix is strictly less than i, the corresponding pixel is black, otherwise, it is white.

For example, if the instruction matrix is

$$\begin{pmatrix} 3 & 1 & 4 & 1 & 5 \\ 9 & 2 & 6 & 5 & 3 \\ 5 & 8 & 9 & 7 & 9 \\ 3 & 2 & 3 & 8 & 4 \\ 6 & 2 & 6 & 4 & 3 \end{pmatrix},$$

then the first 10 images printed by the printer would be:



Note: the last image will then be printed another 990 times by the printer.

Rie has N images that she needs to print but wants to minimise the number of times she needs to operate the printer. Rie is allowed to bring her own input matrices and to use a new matrix for each run of the printer. She doesn't mind if the printer prints unneeded images; she just needs her N images to be printed after she runs the printer several times. Determine the minimum number of times she must operate the printer.

Input

The first line of the input contains a single integer N ($1 \leq N \leq 300$), which is the number of images Rie wants to print.

The next $6N - 1$ lines describe the images Rie wants to print. Each image is described by 5 lines. Each line consists of 5 characters, where each character is # denoting a black pixel, or . denoting a white pixel. There is a blank line between each image. If an image is listed x times in the input, then the printer must print x versions of that image across the several runs of the printer.

Output

Display the minimum number of times Rie has to operate the printer to print the N images.

Sample Input 1

```
1
#.#.#
.....
.....
.....
.....
```

Sample Output 1

```
1
```

Sample Input 2

```
2
#.#.#
.....
.....
.....
.....

#.#.#
.....
.....
.....
.....
```

Sample Output 2

```
1
```

Sample Input 3

```
3
#.#.#
.....
.....
.....
.....

#.#.#
.....
.....
.....
.....

.....
.....
.....
.....
.....
```

Sample Output 3

```
1
```

Sample Input 4	Sample Output 4
3 #.#.# .... #.#.# ....#	2

This page is intentionally left (almost) blank.

Problem I
Incredibly Average
Time limit: 4 seconds

Every year, the Most Incredibly Average Award goes to the array of integers that is the most average. This year, they have decided to give the award to the array based on the following metric:

The judges will first compute the average (specifically, arithmetic mean) of all integers in the array. For example, the average of $[1, 3, 3, 6, 7]$ is $20/5 = 4$. Then, the judges will check all possible contiguous subarrays and compute their average. The score of the array is the number of subarrays that have the same average as the whole array.

In the example above, the subarrays $[3, 3, 6]$ and $[1, 3, 3, 6, 7]$ all have an average of 4, so the score is 2.

Given an array, help the judges by computing the score of the array.

Input

The first line of input contains a single integer N ($1 \leq N \leq 300\,000$), which is the number of elements in the array.

The next line describes the array in order. The line contains N integers $a_1, a_2, \ldots, a_N$ ($1 \leq a_i \leq 1\,000\,000$), which is the array.

Output

Display the score of the array.

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

Sample Input 2	Sample Output 2
4 4 3 2 1	2

Sample Input 3	Sample Output 3
5 1 2 3 4 5	3

This page is intentionally left (almost) blank.

Problem J
Jack and Jill Jotting Notes
Time limit: 4 seconds

Jack and Jill are in a class together and love passing notes. However, they want to keep their message a secret. The secret message they want to send is an N-bit integer, y. Instead of writing down the secret message, they create a list of length $2^N - 1$: $x_1, x_2, \ldots, x_{2^N-1}$ using the formula

$$x_i = y \oplus i,$$

where $\oplus$ is the exclusive-or of two integers.[1] They then shuffle the list into an arbitrary order. The shuffled list is the encrypted message.

For example, if the secret message is 1, then $x_1 = 0, x_2 = 3, x_3 = 2$. They then shuffle this list to give the encrypted message: $[2, 0, 3]$.

Given the encrypted message, what was the original message?

Input

The first line of the input contains a single integer N ($1 \leq N \leq 18$), which is the number of bits in the secret message. That is, the secret message is between 0 and $2^N - 1$, inclusive.

The second line of the input contains $2^N - 1$ integers, x_i ($0 \leq x_i < 2^N$), which are the shuffled values of the encrypted message.

It is guaranteed that the encrypted message was created by using the procedure above.

Output

Display the original message.

Sample Input 1	Sample Output 1
2 2 0 3	1

[1]To calculate the exclusive-or, or XOR, of two numbers: (1) write the number in binary. Then (2) for each bit, the output is 1 if and only if exactly one of the two inputs is 1 for that bit.

This page is intentionally left (almost) blank.

Problem K
Keyboard
Time limit: 15 seconds

Mobile touch-screen devices usually come with onscreen keyboards for text input. To speed up the input, users can draw paths on the keyboard using their fingers. Write a program to find all words in a given dictionary matching the user's input.

Your program needs to work for customised keyboard layouts, such as the one in Figure K.1. A keyboard is a rectangle divided into rows of keys. Keys in a row are rectangles with the same height but varying widths. Heights may vary from row to row. Different keys may be labelled with the same character.

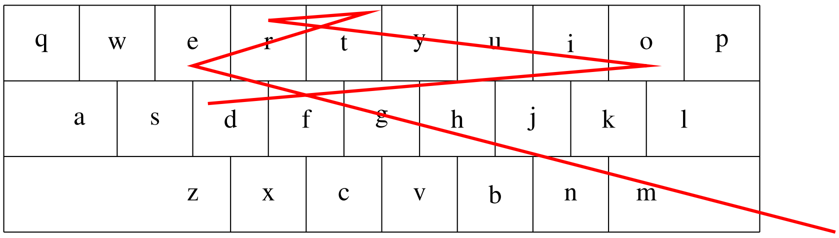


Figure K.1: Keyboard and user's input illustrating sample 1

The user's input is modelled as a polyline in the coordinate system of the keyboard. When traced from its start to its end, the polyline intersects a sequence of keys. Only intersections with the interior of keys count towards this sequence; touching a key border is not enough. Each key that is intersected is "hit". The hit keys define a sequence of key labels.

A word w from the dictionary matches the input if the whole sequence of characters in w is a subsequence of the sequence of hit key labels. A key from the user's input with label k may account for several consecutive occurrences of k in w, that is, for a substring of w containing only the character k.

A *subsequence* of a string is a sequence of characters from that string that is not necessarily contiguous. For example, "cat" is a subsequence of "cant".

You will be given a keyboard layout, the user input, and a dictionary of words. Help us by telling us which keys are on each swipe (that is, on each segment of the polyline) and which words are subsequences of the sequence of key labels.

Input

The input consists of three parts specifying the keyboard layout, the user's input and the dictionary. All numbers in the input are integers.

The first part starts with a line containing the number of rows r in the keyboard r ($1 \le r \le 1000$). Each of the following r lines describes one row, starting with the height h of the keys in this row ($1 \le h \le 1000$) and the number of keys c in this row ($1 \le c \le 1000$). The line is concluded by c key descriptions, each consisting of a key label k (a lower-case letter) and the width of this key w ($1 \le w \le 10^6$). The sum of key widths in each row is the same and is at most 10^6.

If the rows have heights $h_1, h_2, h_3, \ldots$ this means that the first row occupies y-coordinates in the interval $[0, h_1]$, the second row occupies y-coordinates in $[h_1, h_1 + h_2]$, the third row occupies y-coordinates in $[h_1 + h_2, h_1 + h_2 + h_3]$, and so on. A similar fact holds for the x-coordinates of the keys in each row.

The second part of the input starts with a line containing the number of vertices n in the polyline showing the user's input ($2 \le n \le 10^4$). Each of the following n lines contains two numbers x and y giving the respective coordinates of a vertex. Vertices are in the area occupied by the keyboard or close to its border (x- and y-distance each at most 1000). Vertices are ordered as they appear in the polyline. Any two consecutive vertices are different. Any three consecutive vertices may be collinear. There are at most 10^6 keys in the sequence corresponding to the user's input.

The third part of the input starts with a line containing the number of words d in the dictionary ($1 \leq d \leq 10^6$). Each of the following d lines contains one such word (a string of lower-case letters). Any two words are different. The total number of characters of all words in the dictionary is at most $5 \cdot 10^6$.

Output

For each of the $n - 1$ swipes, display the number of distinct keys hit by this swipe and the key labels on those keys, in the order they were hit. Note that if one swipe ends on the interior of a key, it should be output once at the end of one swipe and the beginning of the next swipe even though the key is only hit once.

Then display all words in the dictionary that match the user's input in the order they appear in the input.

Sample Input 1

```
3
10 10 q 10 w 10 e 10 r 10 t 10 y 10 u 10 i 10 o 10 p 10
10 9 a 15 s 10 d 10 f 10 g 10 h 10 j 10 k 10 l 15
10 7 z 30 x 10 c 10 v 10 b 10 n 10 m 20
6
27 13
85 8
35 2
48 1
25 8
110 30
8
ten
fourteen
fen
fern
fin
hint
him
hog
```

Sample Output 1

```
7 dfghuio
6 oiuytr
2 rt
3 tre
9 erdfghjnm
ten
fourteen
fen
fern
fin
him
hog
```

Problem L
Laser Turrets
Time limit: 1 second

The industrial-military machine has finally become sentient, and in order to finally eliminate the human scourge, it has mass-produced laser turrets to completely cover the world. The laser turrets are placed at every positive integer coordinate point on the map (a, b), where $\gcd(a, b) = 1$. Every turret is fixed so that it is always aiming directly away from the point $(0, 0)$ and they are able to destroy any human that crosses in front of that turret. The turret does not destroy the human if it is directly on top of the turret.

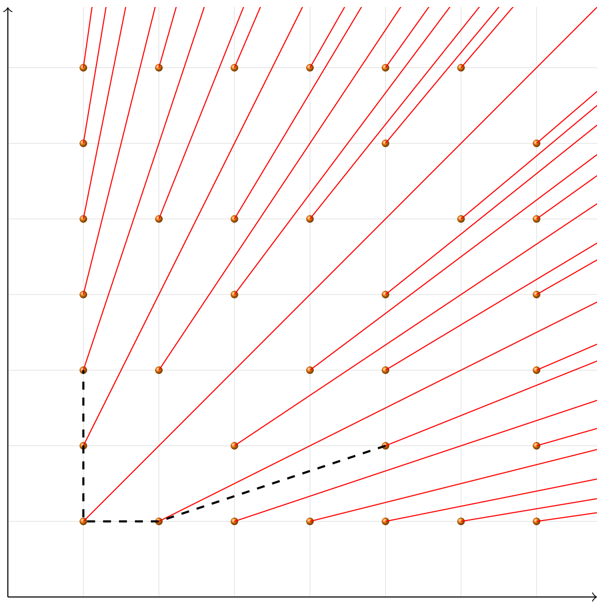


Figure L.1: An optimal path from $(5, 2)$ to $(1, 3)$, which is Sample Input 1.

Chel and other members of the human resistance have found a way to survive in this dystopian world, by living at the base of turrets. They can move from turret to turret in this world as long as they do not travel directly in front of another turret.

Chel has just finished a dangerous mission and needs to get back to the resistance's home base. As Chel's science expert, Chel needs your help to determine a path that travels the minimum distance in order to get back to the home base without being destroyed. The minimum path can be described as a sequence of turrets to visit. No three consecutive turrets in the path's description should be collinear. For example, in the image above, the point $(1, 2)$ should not be included since $(1, 1)$, $(1, 2)$, and $(1, 3)$ are collinear.

Hurry! Determine a minimum path for Chel to take.

Input

The first line contains two integers a ($1 \le a \le 10^{18}$) and b ($1 \le b \le 10^{18}$, $\gcd(a, b) = 1$), denoting that Chel is currently at (a, b).

The second line contains two integers c ($1 \le c \le 10^{18}$) and d ($1 \le d \le 10^{18}$, $\gcd(c, d) = 1$), denoting that the home base is located at (c, d). $(a, b) \ne (c, d)$.

Output

Display a minimum path from Chel's current location back to home base that will not destroy Chel. The path must contain no three consecutive collinear points and must contain no more than 10^5 stops along the way. It is guaranteed that there is some minimum path satisfying these criteria. Any minimum path satisfying these criteria will be considered correct.

Sample Input 1

```
5 2
1 3
```

Sample Output 1

```
5 2
2 1
1 1
1 3
```

Sample Input 2

```
1 3
5 7
```

Sample Output 2

```
1 3
1 2
2 3
5 7
```

Sample Input 3

```
5 2
1 2
```

Sample Output 3

```
5 2
2 1
1 1
1 2
```