

Problem A. Attendance Anxiety

At UFBA, Marina is trying to decide on which days she should attend a demanding review session before the next GruPro selection contest.

For each of the next N days, attending on day i gives her a benefit of a_i , which may even be negative if she is already too exhausted.

If Marina attends on several consecutive days, that forms a single streak. A streak of length L causes a fatigue penalty of L^2 . In other words, if the attended days are partitioned into maximal consecutive streaks of lengths L_1, L_2, \dots, L_k , then her final score is

$$\left(\sum \text{benefit of all attended days}\right) - \left(L_1^2 + L_2^2 + \dots + L_k^2\right).$$

She may skip any days she wants. After all, everyone needs a break sometimes; nobody is made of steel.

Determine the maximum possible final score.

Input

The first line contains an integer N ($1 \leq N \leq 5000$), the number of days.

The second line contains N integers a_1, a_2, \dots, a_N ($-10^9 \leq a_i \leq 10^9$), where a_i is the benefit of attending on day i .

Output

Print a single integer: the maximum possible final score.

Examples

Sample input 1

```
5
4 4 -10 4 4
```

Sample output 1

```
8
```

Explanation of sample 1

In the first sample, the best strategy is to attend on days 1, 2, 4, and 5. This creates two streaks, each of length 2, so the score is

$$4 + 4 + 4 + 4 - 2^2 - 2^2 = 8.$$

Sample input 2

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

Sample output 2

```
9
```

Problem B. Bonfim Bits

During a cultural fair at UFBA, a group of students is preparing souvenir ribbons inspired by the famous *fitinhas do Bonfim*. Each ribbon design is encoded by an integer, and its binary representation tells which ornaments are present: a bit equal to 1 means that ornament is used, while a bit equal to 0 means it is not. Therefore, the total number of ornaments in a design is exactly the number of bits set to 1.

Professor Rubis wants to approve the next design after a given code N , but he is strict about one detail: the new design must contain exactly M ornaments. Among all valid integers strictly greater than N , he wants the smallest possible one, so that the next approved design is as close as possible to the current code.

Help him find that next ribbon code.

Input

The input contains two integers N and M ($0 \leq N \leq 10^{18}$, $1 \leq M \leq 60$).

Output

Output the code of the next approved ribbon design.

Examples

Sample input 1	Sample output 1
10 2	12
Sample input 2	Sample output 2
7 1	8
Sample input 3	Sample output 3
13 3	14

Problem C. Calma, Calabreso

When the former Big Brother Brazil contestant Davi Brito dropped the phrase “Calma, Calabreso”, the internet did what it always does: it turned the moment into an endless stream of edits, reactions, and inside jokes. The meme became so big that it even inspired a Davi Brito lookalike contest at UFBA, which awarded a “kit calabresa” (*lá ele!*) prize.

To celebrate the chaos, his fans assembled a “calabreso panel” from screenshots, reactions, captions, and meme fragments arranged in a matrix. Each cell stores an integer representing the type of content placed in that position.

A rectangular part of the panel is *calm* if it still looks exactly the same after a rotation of 180° . In other words, if you rotate the rectangle by half a turn, every cell must match the cell that moves into its place. So the top-left corner must match the bottom-right corner, the top-right corner must match the bottom-left corner, and so on.

Some parts of the panel are almost symmetric, but not quite. Others are perfectly balanced, as if even the meme itself had decided to calm down.

Your task is to find the maximum possible area of a calm submatrix.

Formally, a submatrix defined by corners (i_1, j_1) and (i_2, j_2) is calm if

$$A_{i,j} = A_{i_1+i_2-i, j_1+j_2-j}$$

for every cell (i, j) inside the submatrix.

Input

The first line contains two integers N and M ($1 \leq N \leq 200$, $1 \leq M \leq 200$).

Each of the next N lines contains M integers, describing the matrix A ($1 \leq A_{i,j} \leq 10^4$).

Output

Output a single integer: the largest possible area of a calm submatrix.

Examples

Sample input 1	Sample output 1
<pre>3 3 1 2 3 4 5 4 3 2 1</pre>	9

Explanation of sample 1

The whole matrix is valid. After a 180° rotation, each cell matches the one in the opposite position, so the answer is $3 \times 3 = 9$.

Sample input 2	Sample output 2
<pre>3 4 1 7 7 1 2 8 9 2 5 6 6 5</pre>	4

Explanation of sample 2

One can choose the first row of the matrix, which has an area of 4. An alternative is the last row, which also has an area of 4.

Sample input 3

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

Sample output 3

```
1
```

Problem D. DJ Sandália’s Electric Trio

Every February, the streets of Salvador surrender to the *Carnaval*, and no one feels it more than DJ Sandália, the legendary selector who rides the oldest *trio elétrico* circling the Campo Grande circuit. The folião crowd shouts song requests all night long, and DJ Sandália has carefully written down, for each of the N songs in the repertoire, an integer “weight” P_i that measures how badly the crowd wants to hear song i .

To keep the party fair, DJ Sandália wants to pick the next song at random so that song i is chosen with probability exactly $P_i/(P_1 + P_2 + \dots + P_N)$.

There is just one problem: the trio’s randomizer is a battered piece of hardware from the 1980s, and the only thing it knows how to do is to draw an integer *uniformly* at random from a given range. It cannot weight anything by itself; every draw it produces is perfectly uniform.

Working only with this uniform randomizer, DJ Sandália came up with the following strategy. First, choose a single integer K and build three arrays A , B and C , each of length N . Then, to pick a song:

- draw x uniformly from 1 to N ;
- draw y uniformly from 0 to K ;
- if $y < C_x$, play song A_x ; otherwise, play song B_x .

Help DJ Sandália choose K and the arrays A , B and C so that this procedure plays song i with probability *exactly* $P_i/(P_1 + \dots + P_N)$, for every song i .

Input

The first line contains a single integer N ($1 \leq N \leq 1000$), the number of songs.

The second line contains N integers P_1, P_2, \dots, P_N ($1 \leq P_i \leq 10^9$), the weight of each song.

Output

On the first line, print a single integer K ($0 \leq K \leq 10^{18}$).

On the second line, print N integers A_1, \dots, A_N ($1 \leq A_i \leq N$).

On the third line, print N integers B_1, \dots, B_N ($1 \leq B_i \leq N$).

On the fourth line, print N integers C_1, \dots, C_N ($0 \leq C_i \leq K$).

Your answer is accepted if, for every song i , the probability that the described procedure plays song i equals $P_i/(P_1 + \dots + P_N)$ exactly. Any valid answer will be accepted.

Examples

Sample input 1	Sample output 1
1	4
5	1
	1
	0

Explanation of sample 1

With a single song, $K = 4$ and the only bucket points to song 1 on both sides, so song 1 is always chosen.

Sample input 2**Sample output 2**

2	3
1 3	1 2
	2 2
	2 3

Explanation of sample 2

Here $K = 3$. With the printed table, $\Pr[1] = \frac{1}{4} = \frac{P_1}{S}$ and $\Pr[2] = \frac{3}{4} = \frac{P_2}{S}$, so it is a valid answer.

Sample input 3**Sample output 3**

3	3
1 1 2	1 2 3
	3 3 3
	3 3 3

Explanation of sample 3

Here $K = 3$. The printed table gives $\Pr[1] = \Pr[2] = \frac{1}{4}$ and $\Pr[3] = \frac{1}{2}$, matching P_i/S . Any other valid table is also accepted.

Problem E. Elevations of Salvador

The city of Salvador, in Bahia, is famously hilly, and its blocks are laid out on a regular grid. We model the whole neighborhood as the infinite integer grid: every grid point has integer coordinates and is a corner where blocks meet.

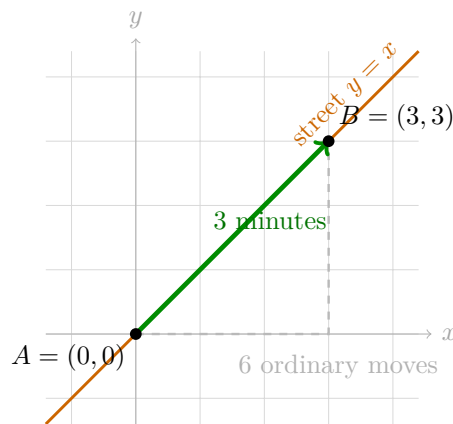
A courier from a UFBA student delivery group starts at corner $A = (a_x, a_y)$ and must reach corner $B = (b_x, b_y)$. In one minute the courier can walk one block to an orthogonally adjacent corner, that is, from (x, y) to any of $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ or $(x, y - 1)$. Each such move costs exactly 1 minute.

Besides the ordinary grid, Salvador also has M special perfectly straight slanted streets, each running at exactly 45 degrees. Each slanted street is described by a pair (k, d) with $d \in \{+1, -1\}$:

- If $d = +1$, the street is the line $y = x + k$. While standing on a grid point of this street, the courier may, in addition to the ordinary moves, step to $(x + 1, y + 1)$ or $(x - 1, y - 1)$ in one minute, staying on the street.
- If $d = -1$, the street is the line $y = -x + k$. While standing on a grid point of this street, the courier may, in addition to the ordinary moves, step to $(x + 1, y - 1)$ or $(x - 1, y + 1)$ in one minute, staying on the street.

These extra slanted steps are available only while the courier is currently on a grid point that belongs to the corresponding slanted street. Every move, ordinary or slanted, costs exactly 1 minute.

For example, suppose $A = (0, 0)$, $B = (3, 3)$, and there is a single slanted street $(k, d) = (0, +1)$, the line $y = x$. The courier can ride this street from A to B in 3 minutes:



Help the group find the minimum number of minutes needed to travel from A to B .

Input

The first line contains four integers a_x , a_y , b_x and b_y ($-10^9 \leq a_x, a_y, b_x, b_y \leq 10^9$), the coordinates of A and B .

The second line contains a single integer M ($0 \leq M \leq 2 \times 10^5$), the number of slanted streets.

Each of the next M lines contains two integers k and d ($-2 \times 10^9 \leq k \leq 2 \times 10^9$, $d \in \{-1, +1\}$), describing one slanted street as defined above.

Output

Output a single integer: the minimum number of minutes needed to travel from A to B .

Examples

Sample input 1

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

Sample output 1

```
3
```

Explanation of sample 1

The slanted street with $k = 0$, $d = +1$ is the line $y = x$, which passes through both $A = (0, 0)$ and $B = (3, 3)$. The courier rides it with three slanted steps $(0, 0) \rightarrow (1, 1) \rightarrow (2, 2) \rightarrow (3, 3)$, for a total of 3 minutes.

Sample input 2

```
0 0 2 3
0
```

Sample output 2

```
5
```

Explanation of sample 2

There are no slanted streets ($M = 0$), so the courier can only move one block at a time. A shortest such walk from $A = (0, 0)$ to $B = (2, 3)$ takes 5 minutes.

Sample input 3

```
0 0 5 1
1
0 1
```

Sample output 3

```
5
```

Explanation of sample 3

The slanted street with $k = 0$, $d = +1$ is the line $y = x$, and $A = (0, 0)$ lies on it while $B = (5, 1)$ does not. The courier takes one slanted step $(0, 0) \rightarrow (1, 1)$ and then walks $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (4, 1) \rightarrow (5, 1)$, for a total of 5 minutes.

Problem F. Fountain at UFBA

The GruPro (Grupo de Programação da UFBA) wants to install a new drinking fountain on the Ondina campus, somewhere among the wooded paths between the *Instituto de Computação* and the *Restaurante Universitário*.

The campus map is an integer grid. There are N benches, and the i -th bench sits at integer coordinates (x_i, y_i) . To be fair to everyone, the students decided the fountain must be placed **exactly halfway between two benches**: that is, at the midpoint of some pair of distinct benches i and j ,

$$\left(\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2} \right).$$

Underground pipes can only reach points with **integer** coordinates, so the chosen midpoint must land exactly on a grid point.

Help the GruPro pick two benches whose midpoint is a valid spot for the fountain, or tell them it is impossible.

Input

The first line contains a single integer N ($2 \leq N \leq 10^5$), the number of benches.

Each of the next N lines contains two integers x_i and y_i ($-10^6 \leq x_i, y_i \leq 10^6$), the coordinates of the i -th bench. No two benches share the same position.

Output

If there is a pair of distinct benches whose midpoint has integer coordinates, print two integers i and j ($1 \leq i, j \leq N$, $i \neq j$): the indices of two such benches. If several pairs work, print any of them.

If no such pair exists, print a single integer -1 .

Examples

Sample input 1	Sample output 1
4 0 0 1 0 2 1 4 2	1 4

Explanation of sample 1

Benches 1 and 4 are at $(0, 0)$ and $(4, 2)$. Their midpoint is $(2, 1)$, which has integer coordinates, so the fountain can be placed there. Benches 1 and 3 would also work; any valid pair is accepted.

Sample input 2	Sample output 2
3 0 0 1 0 0 1	-1

Explanation of sample 2

Checking the three possible pairs, every midpoint has at least one half-integer coordinate. For example, the midpoint of benches 1 and 2 is $(0.5, 0)$. Since no pair gives an integer point, the answer is -1 .

Problem G. GruPro's Golden Challenge

Welcome to GruPro's Golden Challenge! You are a competitive programmer beginning your journey through UFBA's training curriculum. Your programming journey starts with an initial score of X , and ahead of you lies a series of N training challenges, each designed to test and refine your programming abilities in different areas.

However, you have the freedom to choose the order in which you tackle challenges. Each challenge i has two attributes:

- **Score Limit** (P_i): The maximum score you can achieve after completing this challenge;
- **Score Change** (Q_i): The amount your score changes upon completion (can be positive or negative).

When you complete challenge i , your score evolves according to:

$$X \leftarrow \min(P_i, X + Q_i)$$

In other words, your score is adjusted by Q_i points (which can be a gain or a penalty), but your total score cannot exceed the limit P_i for that challenge. Some challenges reward you generously while others penalize careless approaches.

Your mission is to complete all N challenges in the optimal order to **maximize** your final score. Choose wisely, and you might just achieve the ultimate victory!

Input

The first line contains two integers N and X ($1 \leq N \leq 10^5$, $1 \leq X \leq 10^9$).

The next N lines contain two integers P_i and Q_i ($-10^9 \leq P_i, Q_i \leq 10^9$).

Output

Print a single integer: the maximum final score achievable after completing all challenges in the optimal order.

Examples

Sample input 1	Sample output 1
2 5 10 3 8 2	10
Sample input 2	Sample output 2
3 10 15 5 12 -3 20 4	16

Sample input 3

```
4 8
12 2
10 -4
15 5
11 1
```

Sample output 3

```
12
```

Problem H. Heading to Graduation

Marina is in the final stretch of her degree at UFBA. After years of balancing classes, deadlines, and endless cups of coffee, she has only one goal left: **graduate as soon as possible**.

To do that, she must complete all N required disciplines of her curriculum, numbered from 1 to N . At first, her plan seemed simple: whenever a semester begins, she would enroll in every discipline she is already allowed to take. There is no villain in this story, no unexpected internship, no exchange program, and no dramatic plot twist. There is only Marina, her graduation requirements, and the one force capable of turning any peaceful academic plan into chaos: **SIGAA**.

Some disciplines, of course, depend on others. There are M prerequisite relations, and each relation (u, v) means that Marina must complete discipline u before she is allowed to take discipline v . In principle, this should define a sensible curriculum structure. In practice, when SIGAA is involved, Marina would not be entirely surprised if somehow a later discipline ended up demanding an earlier one, which then demanded another one, which eventually demanded the first again.

As if that were not enough, not every discipline is offered every semester. Sometimes departments change plans, sometimes classes do not open, and sometimes SIGAA behaves in ways that students prefer not to question too deeply. There are K temporary restrictions, and each restriction (j, x) means that discipline x is not offered in semester j . So even if Marina is fully ready to take a discipline, SIGAA may simply decide that semester is not the chosen one.

In any semester, Marina may take any number of disciplines, as long as:

- every prerequisite of each chosen discipline was completed in an earlier semester, and
- the discipline is offered in that semester.

Help Marina determine the minimum number of semesters she needs to finish all disciplines, survive the bureaucratic maze, and finally graduate. If graduation is impossible, output -1 and accept that Marina will remain trapped in UFBA forever, eventually becoming yet another legendary campus animal.

Input

The first line contains two integers N and M ($1 \leq N \leq 2 \times 10^5$, $0 \leq M \leq 2 \times 10^5$).

The next M lines each contain two integers u and v ($1 \leq u, v \leq N$, $u \neq v$), meaning that discipline u must be completed before discipline v .

The next line contains one integer K ($0 \leq K \leq 2 \times 10^5$).

The next K lines each contain two integers j and x ($1 \leq j \leq N + K$, $1 \leq x \leq N$), meaning that discipline x is not offered in semester j .

It's guaranteed that all prerequisite pairs are distinct. Additionally, all restriction pairs are distinct.

Output

Output a single integer: the minimum number of semesters needed to complete all disciplines, or -1 if it is impossible.

Examples

Sample input 1

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

Sample output 1

```
4
```

Sample input 2

```
3 3
1 2
2 3
3 1
0
```

Sample output 2

```
-1
```

Problem I. Ivisify

João is about to visit Bahia for the first time. He has already planned everything: walking through Pelourinho, eating a lot of *acarajé*, going to the beach, and learning a few local expressions before he arrives. However, his friends from Salvador warned him that's not enough. According to them, anyone who wants to survive the first few conversations must first pass a small informal test.

They explained one of the rules to him (and the most important one): in some playful situations, a word ending with the letter 'u' is changed to avoid an *unfortunate* rhyme. So, before the trip, João has to practice this transformation correctly.

The transformation is as follows:

- if the word ends with 'u', remove that final 'u' and append 'ivis';
- otherwise, keep the word unchanged.

For example, 'caju' becomes 'cajivis', while 'acarajé' remains 'acarajé'.

Given a word, help João produce the correct Bahia-style version and pass the test.

Input

The input contains a single word S consisting only of lowercase English letters.

It is guaranteed that $1 \leq |S| \leq 100$.

Output

Output the transformed word according to the rule described above.

Examples

Sample input 1	Sample output 1
caju	cajivis
Sample input 2	Sample output 2
acaraje	acaraje
Sample input 3	Sample output 3
urubu	urubivis

Problem J. Joyful Synchronization

It is Carnival in Salvador, and a *bloco afro* is about to parade down the slopes of the Pelourinho with its unmistakable samba-reggae groove.

The bloco is formed by N pairs of percussionists. In the i -th pair, the *surdo* player strikes A_i beats per measure, while the *caixa* player next to them strikes B_i beats per measure. For the whole bloco to lock into a single, hypnotic groove all the way to the Largo do Pelourinho, the maestro wants every pair to produce the **same** total number of beats per measure.

Formally, the maestro picks a target value C and wants $A_i + B_i = C$ to hold for *every* pair i . He is free to choose any C he likes.

To achieve that, he can call rehearsals. In a single rehearsal he picks one drummer (either the *surdo* or the *caixa* of *any* pair) and retunes their pattern to play exactly X beats per measure, where X is any integer with $1 \leq X \leq 10^9$. Notice each rehearsal may use a different value of X .

Help the maestro find the minimum number of rehearsals needed so that all the pairwise sums become equal.

Input

The first line contains a single integer N ($2 \leq N \leq 10^5$), the number of pairs.

The second line contains N integers A_1, A_2, \dots, A_N ($1 \leq A_i \leq 10^6$).

The third line contains N integers B_1, B_2, \dots, B_N ($1 \leq B_i \leq 10^6$).

Output

Output a single integer: the minimum number of rehearsals so that $A_i + B_i$ is the same for every pair.

Examples

Sample input 1	Sample output 1
3 1 2 3 5 4 3	0

Explanation of sample 1

The pairwise sums are $1 + 5 = 6$, $2 + 4 = 6$ and $3 + 3 = 6$. They are already all equal, so the maestro needs no rehearsals.

Sample input 2	Sample output 2
5 2 2 5 8 3 3 3 2 1 9	3

Explanation of sample 2

The current sums are 5, 5, 7, 9, 12. Choosing the target $C = 5$, pairs 1 and 2 are already fine. Each of pairs 3, 4 and 5 can be fixed with a single rehearsal (for example, retune the *surdo* of pair 3 from 5 to 3, the *surdo* of pair 4 from 8 to 4, and the *caixa* of pair 5 from 9 to 2). No choice of C does

better than 3 rehearsals.

Problem K. K-Dendê

At UFBA, a student group is organizing a festival menu and wants to measure the strength of different consecutive stretches of dishes. For each position i , the value A_i represents how much that dish contributes to the overall *dendê power* of the menu. The power of a contiguous segment is the sum of all values inside it.

Everyone already knows how to find the single strongest segment. But for the final ranking, the organizers want more than just the best one.

Among all non-empty contiguous subarrays of A , sort their sums in non-increasing order and determine the K -th one.

Can you help them find the power of the K -th strongest segment?

Input

The first line contains two integers N and K ($1 \leq N \leq 10^5$, $1 \leq K \leq \frac{N(N+1)}{2}$).

The second line contains N integers A_1, A_2, \dots, A_N ($-10^9 \leq A_i \leq 10^9$), where A_i is the *dendê power* contribution of the i -th dish.

Output

Output a single integer: the power of the segment that should appear in the K -th position of the ranking.

Examples

Sample input 1	Sample output 1
3 4 2 -1 3	2
Sample input 2	Sample output 2
4 7 -5 4 -2 3	-1
Sample input 3	Sample output 3
3 5 1 1 1	1

Problem L. Lá ele!

During a GruPro meeting at UFBA, an unfortunate joke started spreading through a friendship network. The network forms a tree with N students, numbered from 1 to N .

Each student is currently in one of two moods:

- 0: Calm, and will let the joke pass;
- 1: Suspicious, and will immediately answer “lá ele!”.

When a story travels through the unique simple path between two students, every suspicious student on that path would react. However, there is an important Bahian rule: once someone says “lá ele!”, everyone in the same consecutive suspicious stretch becomes protected, so no extra “lá ele!” is needed until the path reaches a calm student again.

As the meeting goes on, people may change their mood after hearing new context, remembering an older joke, or deciding that the conversation was actually harmless. Because of that, the current mood of a student is not fixed forever.

You must process two types of operations:

- Update the mood of one student;
- Given two students u and v , compute how many times “lá ele!” is said along the path from u to v .

Input

The first line contains two integers N and Q ($2 \leq N \leq 2 \times 10^5$, $1 \leq Q \leq 2 \times 10^5$).

The second line contains N integers B_1, B_2, \dots, B_N ($0 \leq B_i \leq 1$), where B_i is the initial mood of student i .

Each of the next $N - 1$ lines contains two integers a and b ($1 \leq a, b \leq N$, $a \neq b$), indicating that students a and b are connected. It is guaranteed that these edges form a tree.

Each of the next Q lines describes one operation in one of the following formats:

- 1 u x : Set the mood of student u to x , where $x \in \{0, 1\}$;
- 2 u v : Compute how many times “lá ele!” is said along the path from student u to student v .

Output

For each operation of type 2, output a single integer: the number of times “lá ele!” is said along the queried path.

Examples

Sample input 1

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

Sample output 1

```
3
2
1
1
```

Sample input 2

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

Sample output 2

```
0
1
1
```