The 3rd Universal Cup

Stage 23: Hong Kong

December 28 - December 29, 2024

This problem set should contain 13 problems on 18 numbered pages.

Based on

International Collegiate Programming Contest (ICPC)

Prepared by

Problem A. General Symmetry

Time limit: 2 seconds
Memory limit: 256 megabytes

Let $S = [s_1, s_2, \ldots, s_m]$ be a sequence consisting of m integers. Such a sequence S is called k-symmetric if and only if $|s_i - s_{m-i+1}| \le k$ for all integers i $(1 \le i \le m)$.

You will be given a sequence $A = [a_1, a_2, \ldots, a_n]$ of length n. Your task is to find the length of the longest k-symmetric consecutive subsequence of A centered on each place. Assume that the index range of the corresponding consecutive subsequence is $[l, r]$, it is centered on $\frac{l+r}{2}$.

Input

The first line of the input contains two integers n and k $(2 \le n \le 2 \times 10^5, 0 \le k \le 10^3)$, denoting the length of sequence A and the parameter k.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^3)$, denoting the sequence A.

Output

Print n integers in the first line, the i-th integer $(1 \le i \le n)$ denoting the length of the longest k-symmetric consecutive subsequence of A centered on i.

Print $n - 1$ integers in the second line, the i-th integer $(1 \le i < n)$ denoting the length of the longest k-symmetric consecutive subsequence of A centered on $(i + 0.5)$.

Note that when there is no finding for a fixed center, please print "0" instead.

Examples

standard input	standard output
5 0 1 2 1 2 1	1 3 5 3 1 0 0 0 0
5 1 1 2 1 3 1	1 3 5 3 1 2 2 0 0

Problem B. Defeat the Enemies

Time limit: 3 seconds
Memory limit: 1024 megabytes

You are tasked with defeating n enemies. Each enemy i has a_i health points and b_i armor points. You can use attacks to defeat your enemies. For each of your attacks, you can choose to deal x points of damage to all enemies, where $1 \le x \le k$. Each attack with damage x costs c_x units. You can perform any number of attacks.

When an enemy takes damage, the damage is first absorbed by its armor. When the armor is destroyed, any remaining damage does not carry over to the health. Formally, when enemy x takes y damage, the following happens:

- If $b_x > 0$, b_x is reduced by y.
- Otherwise, a_x is reduced by y.
- The enemy x is considered defeated if at any moment $a_x \le 0$.

Your task is to determine the minimum total cost required to defeat all enemies, and the number of distinct attack strategies (combinations of attacks) that achieve this minimum cost, modulo 998 244 353.

A strategy is an array of integers $x_1, x_2 \dots x_m (1 \le x_i \le k)$, representing the damage dealt in each attack. Two strategies are different if and only if the array x is different.

Input

The first line contains an integer $T (1 \le T \le 1000)$, representing the number of test cases.

For each test case, the first line contains two integers n, m $(1 \le n \le 5 \cdot 10^5, 1 \le m \le 10^4)$, representing the number of enemies, and the maximal health point and armor point.

The following two lines contain two arrays of integers a and b $(1 \le a_i, b_i \le m)$, each of length n, representing the health and armor points of the enemies, respectively.

The following line contains a single integer k $(1 \le k \le 100)$, representing the maximum possible damage per attack.

Then followed by an array of integers c $(1 \le c_i \le 10^9)$, of length k, the i-th integer c_i represents the cost of dealing i damage to all enemies.

It is guaranteed that the sum of n does not exceed $5 \cdot 10^5$, and the sum of m does not exceed 10^4.

Output

For each test case, output one line containing two integers: the minimum total cost to defeat all enemies, and the number of distinct strategies to achieve this cost, modulo 998 244 353.

Example

standard input	standard output
4 5 5 3 5 2 1 2 3 1 3 2 3 3 2 3 4 3 2 2 2 2 2 2 2 3 2 3 3 7 6 5 3 4 6 6 3 4 4 6 4 2 3 5 5 4 2 4 6 7 10 100 38 49 79 66 49 89 21 55 13 23 67 56 26 39 56 16 84 50 92 82 11 6 6 7 8 9 9 9 9 9 9 9	9 1 6 4 18 18 99 44387

Problem C. The Story of Emperor Bie

Time limit: 1 second
Memory limit: 1024 megabytes

Emperor Bie wants to create an array A of length n. He first chooses an initial position p ($1 \le p \le n$), and assigns a positive integer he likes to A_p. He also initializes two variables l and r as $l = r = p$. Then he does $n - 1$ operations, each operation can be any one of the followings:

- Left Expand: This operation can be done if and only if $l > 1$. Emperor Bie chooses an integer k satisfying $0 \le k < A_r$ and assign $A_r - k$ to A_{l-1}. After that, he decreases l by 1.
- Right Expand: This operation can be done if and only if $r < n$. Emperor Bie chooses an integer k satisfying $0 \le k < A_l$ and assign $A_l - k$ to A_{r+1}. After that, he increases r by 1.

Many years later, the Emperor Bie still remembers the array he created, but he has forgotten the initial position p. Please help Emperor Bie to determine which positions could be the initial position p?

Input

The first line contains an integer T ($1 \le T \le 10^5$), the number of test cases.

For each test case, the first line contains one integer n ($1 \le n \le 5 \cdot 10^5$), representing the length of the array.

The following line contains n integers $A_1, A_2 \ldots A_n$ ($1 \le A_i \le 10^9$), representing the array.

It is guaranteed that the sum of n does not exceed $5 \cdot 10^5$, and there exists at least one valid p.

Output

For each test case, output all possible initial positions in increasing order in a single line.

Example

standard input	standard output
3 1 5 2 1 3 3 3 3 3	1 2 1 2 3

Problem D. Master of Both VI

Time limit: 2.5 seconds
Memory limit: 1024 megabytes

*This is the sixth instance of "Master of Both", so it is only natural that the problem is about **Hex**tech.*

In order to seize the Hexgates and complete his glorious evolution, Viktor is attacking Piltover. There are n crossings in Piltover connected by $n-1$ bidirectional roads such that any two crossings can reach each other by using these roads. In other words, Piltover is a tree.

Viktor uses his Hextech Claw to emit Hextech Rays to attack Piltover, while Jayce invented the Hextech Shield to protect the city. Each attack is quantified by a damage value, and each Hextech Shield has a health value. When an attack with damage value a hits a Hextech Shield with health b, the health is reduced to $b-a$. If the health of a Hextech Shield is less than or equal to zero, the Hextech Shield is considered destroyed.

When a Hextech Shield is attacked, it can overload itself to reduce the damage by half, which means the damage could be a real number. For example, if the damage is initially 3 and the shield overloads, the damage will actually be 1.5. However, the Hextech Shield cannot overload during two consecutive attacks on it, as it would cause the Hextech Gemstone to become unstable, resulting in catastrophic consequences.

Under Viktor's relentless attacks, Jayce created numerous Hextech Shields and deployed them at the crossings to block the incoming damage. Once a Hextech Shield is deployed, it will absorb all the attack damage directed at that crossing. Now there are q events happening in order:

- `A` x y z: Viktor emitted a Hextech Ray to attack all crossings on the shortest path from x to y with damage value z.
- `D` x h: Jayce detected a Hextech Shield at crossing x which **hasn't been destroyed** yet. Due to the chaos on the battlefield, Jayce cannot remember when the Hextech Shield was placed. He only remembers the Hextech Shield initially has a health value of h. You should help him calculate the maximum number of attacks it has taken to help Jayce infer the reliability of this Hextech Shield.

The defense must not fail, so please help Jayce finish the calculations.

Input

The first line contains two integers n, q ($1 \le n \le 5 \cdot 10^5, 1 \le q \le 3 \cdot 10^5$), denoting the number of crossings and the number of events.

Each of the following $n-1$ lines contains two integers u, v ($1 \le u, v \le n$), denoting a bidirectional edge connecting crossings u and v.

The i-th of the following q lines contains an event, starting with a character o_i ($o_i \in \{\texttt{A}, \texttt{D}\}$). If $o_i = \texttt{A}$, then three integers x_i, y_i, z_i ($1 \le x_i, y_i \le n, 1 \le z_i \le 370$) follow, describing an attack event. Otherwise, two integers x_i, h_i ($1 \le x_i \le n, 1 \le h_i \le 10^9$) follow, describing a detection event.

Output

For each event of the detection type, output the answer in a single line.

Example

standard input	standard output
5 7 1 2 1 3 2 4 2 5 A 1 4 2 A 3 5 2 D 1 4 D 2 3 D 2 1 A 5 5 10 D 5 100	2 1 0 2

Problem E. Concave Hull

Time limit: 3 seconds
Memory limit: 1024 megabytes

You are given n points on the plane; the coordinates of the i-th point are (x_i, y_i).

A Concave Hull is a simple polygon (meaning it has no self-intersections) such that the set of vertices is a non-empty subset of the given n points, and all the n points lie inside or on the boundary of the polygon. Exactly one of the interior angles of the polygon is greater than π, and all the other angles are less than π.

Calculate twice the sum of the area of all the Concave Hulls of the given set of points. Since the answer might be large, output it modulo $10^9 + 7$.

Input

The first line contains one integer n ($3 \le n \le 2 \cdot 10^3$), denoting the number of points.

The i-th of the following n lines contains two integers x_i, y_i ($0 \le x_i, y_i \le 10^9$), denoting the coordinates of the i-th point.

It is guaranteed that the n points are pairwise distinct, and no three points are collinear.

Output

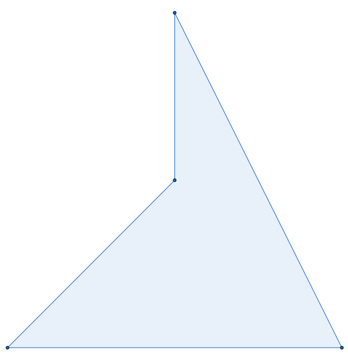
Output a single integer in one line, denoting twice the sum of the area of the Concave Hulls modulo $10^9 + 7$.

Examples

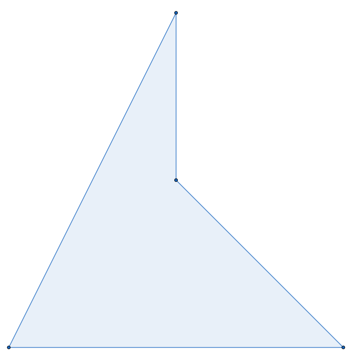
standard input	standard output
4 0 0 2 0 1 2 1 1	8
15 3442 3341 3136 3120 3228 3113 3143 2981 3050 3052 2970 2973 2964 3011 2921 2927 2844 2715 2655 2661 2666 2637 2755 2731 2657 2684 2662 2629 2542 2508	23993862

Note

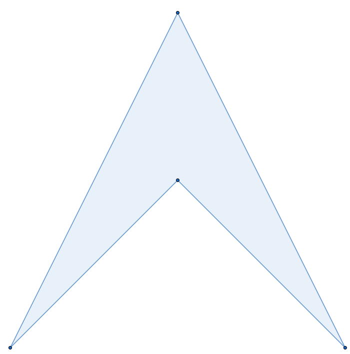
For the first sample, there are three Concave Hulls.



the area is 1.5



the area is 1.5



the area is 1

So the total area is 4, and the answer is $4 \times 2 = 8$.

Problem F. Money Game 2

Time limit: 4 seconds
Memory limit: 1024 megabytes

Putata and Budada are organizing a game with n players sitting in a circle. The players are numbered from 0 to $n-1$, and the i-th player is adjacent to the $((i-1) \bmod n)$-th and $((i+1) \bmod n)$-th player. Player i has a_i deposits in the beginning, which is an integer.

During each round of the game, the following happens: Putata and Budada will choose a player x who hasn't been chosen before, and the player x will give half of his deposit (rounded down to an integer) to one of his adjacent players. Notice that each player can share his deposit only once.

Let $f(i)$ be the maximum possible deposits that the player numbered i will have after some (possibly zero) number of rounds. Please, for each $0 \le i < n$, calculate $f(i)$. Notice that the answers are calculated independently for different values of i.

Input

The input contains multiple test cases. The first line contains an integer T ($1 \le T \le 5 \cdot 10^5$), denoting the number of test cases.

For each test case, the first line contains an integer n ($1 \le n \le 5 \cdot 10^5$), denoting the number of players.

The second line contains n integers, the i-th integer a_i ($0 \le a_i \le 10^9$), denoting the initial deposit of the i-th player.

It is guaranteed that the sum of n does not exceed $5 \cdot 10^5$.

Output

For each test case, output n integers in one line, denoting $f(0), f(1), \ldots, f(n-1)$.

Examples

standard input	standard output
3 5 2 1 4 3 5 5 2 1 3 1 2 1 1000000000	6 5 7 8 8 4 4 5 4 4 1000000000
1 10 8 15 18 15 13 4 14 4 17 5	30 37 41 39 34 27 29 26 31 27

Problem G. Yelkrab

Time limit: 2 seconds
Memory limit: 1024 megabytes

The International Collegiate Piggy Contest is about to start! This contest features different tracks, each with specific requirements for the number of participants per team. For the k-th track, the number of participants in each team must be **exactly** k.

There are n piggies in Pigetown University, and they want to participate in a way that maximizes the rating of their University. The i-th piggy's name is s_i, which is a string consisting of lowercase English letters. The rating of the team is the length of the longest common prefix† of their name. The rating of the University is the sum of ratings of all the teams sent by this University. Each piggy can only participate in at most one team.

Let $f(i,j)$ be the maximum rating of the Pigetown University if the Pigetown University only sends teams consisting of the first i piggies to participate in the j-th track, where each team consists of exactly j piggies. Please, for each $1 \le i \le n$, calculate $\bigoplus_{j=1}^{i}(f(i,j) \times j)$, where $\oplus$ denotes bitwise exclusive or operation.

†: The length of the longest common prefix of m strings $t_1, t_2, \ldots, t_m$ is the largest non-negative integer p, where $p \le \min(|t_1|, |t_2|, \ldots, |t_m|)$ and for all $1 \le e \le p, 1 \le i, j \le m$, $t_{i,e} = t_{j,e}$.

Input

The input contains multiple test cases. The first line contains an integer T ($1 \le T \le 5 \cdot 10^5$).

For each test case, the first line contains an integer n ($1 \le n \le 5 \cdot 10^5$).

The i-th of the following n lines contains a string s_i ($1 \le |s_i| \le 10^6$), denoting the name of the i-th piggy. It is guaranteed that the string only consists of lowercase English letters.

It is guaranteed that the sum of n does not exceed $5 \cdot 10^5$, and the sum of the names of piggies in all test cases does not exceed 10^6.

Output

For each test case, output n integers in one line, separated by spaces. The i-th integer is $\bigoplus_{j=1}^{i}(f(i,j) \times j)$.

Example

standard input	standard output
2 5 aa ab ab ac d 1 aaaaa	2 6 1 9 8 5

Problem H. Mah-jong

Time limit: 3 seconds
Memory limit: 1024 megabytes

A positive integer multiset s is a "*Pong*" if $s = \{x, x, x\}$ for some positive integer x.

A positive integer multiset s is a "*Chow*" if $s = \{x, x+1, x+2\}$ for some positive integer x.

A positive integer multiset is a "*Mahjong*" if it can be divided into some (possibly zero) "*Pong*"s and some (possibly zero) "*Chow*"s. Note that the definition is **different** from the original Mahjong.

Now you are given n integers $a_1, a_2 \ldots a_n$. Your task is to calculate the number of intervals $[l, r] (1 \le l \le r \le n)$ such that the multiset $\{a_l, a_{l+1} \ldots a_r\}$ is a "*Mahjong*".

Input

The input contains multiple test cases. The first line contains one integer T $(1 \le T \le 100)$, representing the number of test cases.

For each test case, the first line contains one integer n $(1 \le n \le 10^5)$, representing the number of integers. The following line contains n integers $a_1, a_2 \ldots a_n$ $(1 \le a_i \le 8)$.

It is guaranteed that the sum of n does not exceed 10^5.

Output

For each test case, output one integer, representing the answer.

Example

standard input	standard output
5 4 1 1 1 1 6 1 2 3 1 2 3 7 6 5 8 7 6 3 2 8 1 2 1 2 1 2 1 3 9 2 2 4 4 1 1 1 3 3	2 5 1 3 2

Problem I. Ma Meilleure Ennemie

Time limit: 1.5 seconds
Memory limit: 1024 megabytes

"Can we just pretend like it's the first time?"

Ekko has just invented the Z-Drive, which allows him to travel back in time. Heimerdinger developed this device to send Ekko back to the correct Universe.

However, due to the spatio-temporal distortion caused by the anomalies, there are n Ekkos that have been transported to this Universe. It is hard to find where to send them back, so Heimerdinger invented a method to determine which Universe these Ekkos belong to. There are m Universes, and the method works as follows:

- The Ekkos are numbered from 0 to $n-1$ and are divided into several groups. Initially, all the Ekkos are in the same group.
- Heimerdinger chooses a group and tries to determine which Universe he should send the Ekkos to.
- Heimerdinger chooses a **good** integer x from $[1, m]$, where an integer x is considered good if and only if, for each Ekko i in this group, the Ekko numbered $(i + x) \bmod n$ is also in this group. For example, if the chosen group contains Ekko $0, 2$, and $n = 4$, then $x = 3$ is not good, while $x = 2$ and $x = 4$ are good.
- Then Heimerdinger divides Ekkos in this group into several *temporary groups*. The division ensures that for each Ekko i in any of the *temporary groups*, the Ekko numbered $(i + x) \bmod n$ is in the same *temporary group* as Ekko i. Heimerdinger will choose the division that maximizes the number of *temporary groups* among all possible divisions when the good integer is x. For example, when $n = 4$ and the group contains $\{0, 1, 2, 3\}$, choosing $x = 2$ will divide this group into two *temporary groups* $\{0, 2\}$ and $\{1, 3\}$.
- If there is only a single *temporary group*, then all the Ekkos in this group are sent back to Universe x. Otherwise, each *temporary group* becomes a new group, and the process repeats with the new groups formed.

Even though this method of discrimination has some minor issues, such as sending multiple Ekkos back to the same universe, there is no time to address it. Given n, m, you need to calculate the number of different results produced by this method of discrimination, modulo 998 244 353. Two results are considered different if and only if some Ekko is sent back into different Universes in the two results.

Input

The first line contains two integers n, m ($1 \le n, m \le 10^{18}$), denoting the number of Ekkos and the number of universes.

Output

Output one integer in one line, denoting the answer, modulo 998 244 353.

Examples

standard input	standard output
4 4	6
2338 1470	18530141

Problem J. Reconstruction

Time limit: 3 seconds
Memory limit: 1024 megabytes

Pig100ton has a tree T_1 with n vertices and an array a of length n whose elements are initially equal to 0. He can construct a new tree T_2 from T_1 by performing the following operations for n times:

- Choose an arbitrary vertex x that is not deleted in T_1. Let its father in T_2 be a_x (if $a_x = 0$, then let x be the root of T_2).
- For all vertices y that can be reached from x by edges in T_1, assign x to a_y.
- Delete vertex x and the edges adjacent to x in T_1.

Pig100ton has another tree T of n vertices. For each $1 \le u \le n$, he wants to know whether T is a possible tree that can be constructed by him from T_1 if T is rooted at vertex u. Please help him find the answers.

Input

The first line contains a single integer n ($1 \le n \le 5 \cdot 10^5$), denoting the number of vertices in the tree T_1.

Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n, u \ne v$), representing an undirected edge from u to v in T_1. It is guaranteed that the given edges form a tree.

Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n, u \ne v$), representing an undirected edge from u to v in T. It is guaranteed that the given edges form a tree.

Output

Output a string of length n in a single line. The i-th character is '1' if T is a possible tree that can be constructed by him from T_1 when T is rooted at vertex i, or '0' otherwise.

Examples

standard input	standard output
3 1 2 2 3 2 1 1 3	001
6 1 3 3 4 3 6 4 5 5 2 1 3 1 4 4 5 5 2 3 6	010110

Problem K. LR String

Time limit: 2 seconds
Memory limit: 1024 megabytes

Gamenjoy has a favorite string consisting of only "L"s and "R"s.

However, Gamenjoy's naughty brother, Gamanyong, has modified the string. Formally, he does several (possibly zero) operations where each operation can be one of the following:

- Choose a character "L" in the string which is not the first character and delete the character immediately to its left.
- Choose a character "R" in the string which is not the last character and delete the character immediately to its right.

When Gamenjoy returns home, he can't distinguish which one is his favorite string anymore. Luckily, he remembers what the string looked like before his brother modified it. Gamenjoy finds q strings at home and asks their father, Oibuth, whether each string could possibly be his favorite string. Oibuth now turns to you for help. Please help him answer Gamenjoy's questions.

Input

The first line contains one integer T ($1 \le T \le 10^5$), representing the number of test cases.

For each test case, the first line contains a string s ($1 \le |s| \le 5 \cdot 10^5$), consisting of only "L"s and "R"s, representing the original string.

The second line contains an integer q ($1 \le q \le 5 \cdot 10^5$), representing the number of strings in Gamenjoy's home.

The i-th of the following q lines contains a string t_i ($1 \le |t_i| \le |s|$) consisting of only "L"s and "R"s, representing a string in Gamenjoy's home.

It is guaranteed that the sum of $|s|$, the sum of q, and the sum of $|t_i|$ over all test cases do not exceed 10^6.

Output

For each string in Gamenjoy's home, output "YES" if the string could be his favorite string. Otherwise, output "NO".

Example

standard input	standard output
2 RRLLRRLL 4 LLLLL LLR LRLR R RLLLLLL 3 LLLLL RL RRL	NO YES NO YES YES YES NO

Problem L. Flipping Paths

Time limit: 1 second
Memory limit: 1024 megabytes

Grammy has a rectangular sheet of paper of size $n \times m$ divided into unit square cells. Initially, some cells are black, and others are white. Grammy has to paint all the cells the same color. In order to achieve that, Grammy can perform the following operation at most 400 times:

- Place the pencil at the top-left cell of the sheet (assume this cell has coordinates $(1, 1)$), and draw a path to the bottom-right cell of the sheet (the cell with coordinates (n, m)). The path must go along the cells, moving only down or right. Specifically, if the pencil is at (x, y), it can move either down to $(x + 1, y)$ or right to $(x, y + 1)$. Grammy must ensure the pencil remains within the sheet.
- After that, change the color of all cells located on the path (that is, paint all white cells on the path black, and all black cells on the path white).
- Finally, erase the drawn path, and proceed to the next operation (if necessary).

Please help Grammy find a way to paint all the cells the same color in at most 400 operations, or state that such a way does not exist.

Input

There are multiple test cases.

The first line contains a single integer T $(1 \le T \le 500)$, denoting the number of test cases. For each test case:

The first line contains two integers n, m $(1 \le n, m \le 200, 2 \le nm)$, denoting the number of rows and columns of the grid.

Each of the next n lines contains m characters c_{ij} ($c_{ij} \in \{$‘`W`‘, ‘`B`‘$\}$), denoting the initial color of each cell. ‘`W`‘ stands for black (“wakuda” in Chewa) and ‘`B`‘ stands for white (“biancu” in Corsican).

It is guaranteed that neither the sum of n nor the sum of m exceeds 1000.

Output

For each test case:

If the solution does not exist, output “`NO`” on a single line.

Otherwise, output “`YES`” on the first line, then output a single integer k $(0 \le k \le 400)$ on the second line, denoting the number of operations used. Finally, output k lines describing the paths used in the operations.

Each path should be described as a string of moves (‘`R`‘ for right and ‘`D`‘ for down). You should make sure that those moves actually form a path from $(1, 1)$ to (n, m).

Example

standard input	standard output
4 3 3 WBB BWB BBW 1 5 WWWWW 2 2 BB BB 4 1 W B B W	YES 2 RRDD DDRR YES 0 YES 0 NO

Problem M. Godzilla

Time limit: 5 seconds
Memory limit: 1024 megabytes

Godzilla, the ancient Titan monster, is on the way to Bytetown!

Bytetown has $n \times m$ blocks, whose top view is a grid of n rows and m columns. The rows and columns of the grid are numbered from 1 to n and 1 to m, respectively.

Godzilla will visit each block exactly once. Assume Godzilla is at block (i, j) now, then he can choose to do nothing or consume $e(i, j)$ units of energy to make an attack in one of the following two ways. Either of them will cause $d(i, j)$ units damage to Bytetown:

- Cast "Horizontal Atomic Breath" to hit every block located at the i-th row.
- Cast "Vertical Atomic Breath" to hit every block located at the j-th column.

Note that Godzilla won't make multiple attacks in the same block.

Godzilla is cruel but fair. So each block will be hit by "Horizontal Atomic Breath" exactly once and will also be hit by "Vertical Atomic Breath" exactly once. Find the maximum possible total damage Godzilla can cause when the total energy consumed by Godzilla modulo 4 is k. You can safely assume that Godzilla always has enough energy.

Input

The first line of the input contains two integers n and m ($2 \le n, m \le 75$), denoting the number of rows and columns.

In the next n lines, the i-th line contains m integers $d(i,1), d(i,2), \ldots, d(i,m)$. ($1 \le d(i,j) \le 10^7$)

In the next n lines, the i-th line contains m integers $e(i,1), e(i,2), \ldots, e(i,m)$. ($0 \le e(i,j) \le 3$)

Output

Print four lines. In the i-th ($1 \le i \le 4$) line, print a single integer, denoting the maximum possible total units of damage that can be caused when the total units of energy consumed by Godzilla modulo 4 is $(i - 1)$. Note that when it is impossible, please print "`-1`" instead.

Examples

standard input	standard output
2 2 1 2 3 4 2 1 0 3	-1 -1 10 -1
3 3 1 2 3 4 5 6 7 8 9 2 1 0 0 3 2 1 2 1	35 38 37 36