C. Dungeons and Candies

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

During the loading of the game "Dungeons and Candies" you are required to get descriptions of k levels from the server. Each description is a map of an $n \times m$ checkered rectangular field. Some cells of the field contain candies (each cell has at most one candy). An empty cell is denoted as "." on the map, but if a cell has a candy, it is denoted as a letter of the English alphabet. A level may contain identical candies, in this case the letters in the corresponding cells of the map will be the same.

When you transmit information via a network, you want to minimize traffic — the total size of the transferred data. The levels can be transmitted in any order. There are two ways to transmit the current level A:

1. You can transmit the whole level A. Then you need to transmit $n \cdot m$ bytes via the network.
2. You can transmit the difference between level A and some previously transmitted level B (if it exists); this operation requires to transmit $d_{A,B} \cdot w$ bytes, where $d_{A,B}$ is the number of cells of the field that are different for A and B, and w is a constant. Note, that you should compare only the corresponding cells of levels A and B to calculate $d_{A,B}$. You cannot transform the maps of levels, i.e. rotate or shift them relatively to each other.

Your task is to find a way to transfer all the k levels and minimize the traffic.

Input

The first line contains four integers n, m, k, w ($1 \le n, m \le 10$; $1 \le k, w \le 1000$). Then follows the description of k levels. Each level is described by n lines, each line contains m characters. Each character is either a letter of the English alphabet or a dot ("."). Please note that the case of the letters matters.

Output

In the first line print the required minimum number of transferred bytes.

Then print k pairs of integers $x_1, y_1, x_2, y_2, \ldots, x_k, y_k$, describing the way to transfer levels. Pair x_i, y_i means that level x_i needs to be transferred by way y_i. If y_i equals 0, that means that

Zepto Code Rush 2014

Finished

Practice

the level must be transferred using the first way, otherwise y_i must be equal to the number of a previously transferred level. It means that you will transfer the difference between levels y_i and x_i to transfer level x_i. Print the pairs in the order of transferring levels. The levels are numbered 1 through k in the order they follow in the input.

If there are multiple optimal solutions, you can print any of them.

Sample test(s)

input

```
2 3 3 2
A.A
...
A.a
..C
X.Y
...
```

output

```
14
1 0
2 1
3 1
```

input

```
1 1 4 1
A
.
B
.
```

output

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

input

```
1 3 5 2
ABA
BBB
BBA
BAB
ABB
```

output

```
11
1 0
3 1
2 3
4 2
5 1
```

ZOJ Problem Set - 3780

Paint the Grid Again

Time Limit: 2 Seconds Memory Limit: 65536 KB

Leo has a grid with $N \times N$ cells. He wants to paint each cell with a specific color (either black or white).

Leo has a magical brush which can paint any row with black color, or any column with white color. Each time he uses the brush, the previous color of cells will be covered by the new color. Since the magic of the brush is limited, each row and each column can only be painted at most once. The cells were painted in some other color (neither black nor white) initially.

Please write a program to find out the way to paint the grid.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer N ($1 <= N <= 500$). Then N lines follow. Each line contains a string with N characters. Each character is either 'X' (black) or 'O' (white) indicates the color of the cells should be painted to, after Leo finished his painting.

Output

For each test case, output "No solution" if it is impossible to find a way to paint the grid.

Otherwise, output the solution with minimum number of painting operations. Each operation is either "R#" (paint in a row) or "C#" (paint in a column), "#" is the index (1-based) of the row/column. Use exactly one space to separate each operation.

Among all possible solutions, you should choose the lexicographically smallest one. A solution X is lexicographically smaller than Y if there exists an integer k, the first k - 1 operations of X and Y are the same. The k-th operation of X is smaller than the k-th in Y. The operation in a column is always smaller than the operation in a row. If two operations have the same type, the one with smaller index of row/column is the lexicographically smaller one.

Sample Input

```
2
2
XX
OX
2
XO
OX
```

Sample Output

```
R2 C1 R1
No solution
```

Author: **YU, Xiaoyao**
Source: **The 11th Zhejiang Provincial Collegiate Programming Contest**

Submit Status

D. Cunning Gena

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A boy named Gena really wants to get to the "Russian Code Cup" finals, or at least get a t-shirt. But the offered problems are too complex, so he made an arrangement with his n friends that they will solve the problems for him.

The participants are offered m problems on the contest. For each friend, Gena knows what problems he can solve. But Gena's friends won't agree to help Gena for nothing: the i-th friend asks Gena x_i rubles for his help **in solving all the problems** he can. Also, the friend agreed to write a code for Gena only if Gena's computer is connected to at least k_i monitors, each monitor costs b rubles.

Gena is careful with money, so he wants to spend as little money as possible to solve all the problems. Help Gena, tell him how to spend the smallest possible amount of money. Initially, there's no monitors connected to Gena's computer.

Input

The first line contains three integers n, m and b ($1 \le n \le 100$; $1 \le m \le 20$; $1 \le b \le 10^9$) — the number of Gena's friends, the number of problems and the cost of a single monitor.

The following $2n$ lines describe the friends. Lines number $2i$ and $(2i + 1)$ contain the information about the i-th friend. The $2i$-th line contains three integers x_i, k_i and m_i ($1 \le x_i \le 10^9$; $1 \le k_i \le 10^9$; $1 \le m_i \le m$) — the desired amount of money, monitors and the number of problems the friend can solve. The $(2i + 1)$-th line contains m_i distinct positive integers — the numbers of problems that the i-th friend can solve. The problems are numbered from 1 to m.

Output

Print the minimum amount of money Gena needs to spend to solve all the problems. Or print -1, if this cannot be achieved.

Sample test(s)

input

```
2 2 1
100 1 1
2
100 2 1
1
```

output

```
202
```

input

```
3 2 5
100 1 1
1
100 1 1
2
200 1 2
1 2
```

output

```
205
```

input

```
1 2 1
1 1 1
1
```

output

```
-1
```

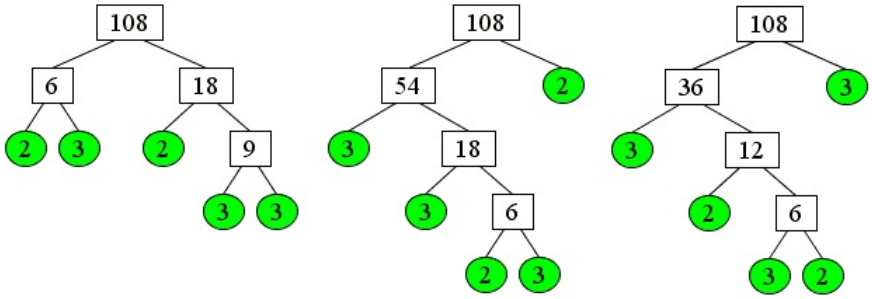
ZOJ Problem Set - 3405

Counting Factor Trees

Time Limit: 2 Seconds Memory Limit: 65536 KB

Factoring, i.e., listing all the prime factors, of an integer is a useful skill that often helps to solve math problems. For example, one of the ways to find the GCD (Greatest Common Divisor) or LCM (Least Common Multiple) of two integers is by listing all their prime factors. The GCD is then the product of all the common factors; the LCM is the product of all the remaining ones.

The *Factor Tree* is a tool for finding such prime factorizations. The figure below demonstrates three factor trees of 108. At the beginning a root with a number is given, say N, which is to be factored. Then, the root is factored into two children N_1 and N_2 such that $N = N_1 \times N_2$ ($N_1 \geq 2$, $N_2 \geq 2$). Note that N_1 and N_2 need not be prime. The same factoring process continues until all the leaves are prime.



While the prime factorization is unique, the factor tree reflects the order in which the factors were found, and is by no means unique. So, how many factor trees of a number are there?

Input

There are no more than 10000 cases. A line containing an integer N ($2 \leq N \leq 1000000000$) is given for each case.

Output

Print the number of factor trees of N in a line for each case. The answer will be fit in a signed 64-bit integer.

Sample Input

```
12
108
642485760
```

Sample Output

```
6
140
9637611984000
```

Author: **GAO, Yuan**
Contest: **ZOJ Monthly, September 2010**

Submit Status

D. Giving Awards

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The employees of the R1 company often spend time together: they watch football, they go camping, they solve contests. So, it's no big deal that sometimes someone pays for someone else.

Today is the day of giving out money rewards. The R1 company CEO will invite employees into his office one by one, rewarding each one for the hard work this month. The CEO knows who owes money to whom. And he also understands that if he invites person x to his office for a reward, and then immediately invite person y, who has lent some money to person x, then they can meet. Of course, in such a situation, the joy of person x from his brand new money reward will be much less. Therefore, the R1 CEO decided to invite the staff in such an order that the described situation will not happen for any pair of employees invited one after another.

However, there are a lot of employees in the company, and the CEO doesn't have a lot of time. Therefore, the task has been assigned to you. Given the debt relationships between all the employees, determine in which order they should be invited to the office of the R1 company CEO, or determine that the described order does not exist.

Input

The first line contains space-separated integers n and m $(2 \le n \le 3 \cdot 10^4; 1 \le m \le min(10^5, \frac{n(n-1)}{2}))$ — the number of employees in R1 and the number of debt relations. Each of the following m lines contains two space-separated integers a_i, b_i $(1 \le a_i, b_i \le n; a_i \ne b_i)$, these integers indicate that the person number a_i owes money to a person a number b_i. Assume that all the employees are numbered from 1 to n.

It is guaranteed that each pair of people p, q is mentioned in the input data at most once. In particular, the input data will not contain pairs p, q and q, p simultaneously.

Output

Print -1 if the described order does not exist. Otherwise, print the permutation of n distinct integers. The first number should denote the number of the person who goes to the CEO office first, the second number denote the person who goes second and so on.

If there are multiple correct orders, you are allowed to print any of them.

Sample test(s)

input
2 1 1 2
output
2 1

input
3 3 1 2 2 3 3 1
output
2 1 3

E. Cardboard Box

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Everyone who has played Cut the Rope knows full well how the gameplay is organized. All levels in the game are divided into boxes. Initially only one box with some levels is available. Player should complete levels to earn stars, collecting stars opens new box with levels.

Imagine that you are playing Cut the Rope for the first time. Currently you have only the levels of the first box (by the way, it is called "Cardboard Box"). Each level is characterized by two integers: a_i — how long it takes to complete the level for one star, b_i — how long it takes to complete the level for two stars ($a_i < b_i$).

You want to open the next box as quickly as possible. So, you need to earn at least w stars. How do make it happen? Note that the level can be passed only once: either for one star or for two. You do not necessarily need to pass all the levels.

Input

The first line contains two integers n and w ($1 \le n \le 3 \cdot 10^5$; $1 \le w \le 2n$) — the number of levels in the first box and the number of stars you need to open another box. Each of the following n lines contains two integers a_i and b_i ($1 \le a_i < b_i \le 10^9$) — the attributes of the i-th level.

Output

In the first line print integer t — the minimum time you need to open the next box.

In the next line, print n digits without spaces — the description of the optimal scenario:

- if you need to pass the i-th level for one star, the i-th digit should equal 1;
- if you need to pass the i-th level for two stars, the i-th digit should equal 2;
- if you do not need to pass the i-th level at all, the i-th digit should equal 0.

Sample test(s)

input
2 3 1 2 1 2
output
3 12

input
5 3 10 20 5 10 10 20 6 9 25 30
output
14 01020

Note

In the first test sample, answer 21 is also assumed correct.

ZOJ Problem Set - 3656

Bit Magic

Time Limit: 8 Seconds Memory Limit: 32768 KB

Yesterday, my teacher taught me about bit operators: and (&), or (|), xor (^). I generated a number table a[N], and wrote a program to calculate the matrix table b[N][N] using three kinds of bit operator. I thought my achievement would get teacher's attention.

The key function is the code showed below.

```
void calculate(int a[N], int b[N][N]) {
        for (int i = 0; i < N; ++i) {
                for (int j = 0; j < N; ++j) {
                        if (i == j) b[i][j] = 0;
                        else if (i % 2 == 1 && j % 2 == 1) b[i][j] = a[i] | a[j];
                        else if (i % 2 == 0 && j % 2 == 0) b[i][j] = a[i] & a[j];
                        else b[i][j] = a[i] ^ a[j];
                }
        }
}
```

There is no doubt that my teacher raised lots of interests in my work and was surprised to my talented programming skills. After deeply thinking, he came up with another problem: if we have the matrix table b[N][N] at first, can you check whether corresponding number table a[N] exists?

Input

There are multiple test cases.

For each test case, the first line contains an integer N, indicating the size of the matrix. (1 ≤ N ≤ 500).

The next N lines, each line contains N integers, the jth integer in ith line indicating the element b[i][j] of matrix. (0 ≤ b[i][j] ≤ 2 ^ 31 - 1)

Output

For each test case, output "YES" if corresponding number table a[N] exists; otherwise output "NO".

Sample Input

```
2
0 4
4 0
3
0 1 24
1 0 86
24 86 0
```

Sample Output

```
YES
NO
```

Contest: **The 2012 ACM-ICPC Asia Changchun Regional Contest**

Submit Status

A. Coder

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Iahub likes chess very much. He even invented a new chess piece named Coder. A Coder can move (and attack) one square horizontally or vertically. More precisely, if the Coder is located at position (x, y), he can move to (or attack) positions $(x + 1, y)$, $(x–1, y)$, $(x, y + 1)$ and $(x, y–1)$.

Iahub wants to know how many Coders can be placed on an $n \times n$ chessboard, so that no Coder attacks any other Coder.

Input

The first line contains an integer n $(1 \leq n \leq 1000)$.

Output

On the first line print an integer, the maximum number of Coders that can be placed on the chessboard.

On each of the next n lines print n characters, describing the configuration of the Coders. For an empty cell print an '.', and for a Coder print a 'C'.

If there are multiple correct answers, you can print any.

Sample test(s)

input
2

output
2 C. .C

C. Barcode

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You've got an $n \times m$ pixel picture. Each pixel can be white or black. Your task is to change the colors of as few pixels as possible to obtain a barcode picture.

A picture is a barcode if the following conditions are fulfilled:

- All pixels in each column are of the same color.
- The width of each monochrome vertical line is at least x and at most y pixels. In other words, if we group all neighbouring columns of the pixels with equal color, the size of each group can not be less than x or greater than y.

Input

The first line contains four space-separated integers n, m, x and y ($1 \le n, m, x, y \le 1000$; $x \le y$).

Then follow n lines, describing the original image. Each of these lines contains exactly m characters. Character "`.`" represents a white pixel and "#" represents a black pixel. The picture description doesn't have any other characters besides "`.`" and "#".

Output

In the first line print the minimum number of pixels to repaint. It is guaranteed that the answer exists.

Sample test(s)

input
6 5 1 2 ##.#. .###. ###.. #...# .##.# ###..
output
11

input
2 5 1 1 ##### .....
output
5

Note

In the first test sample the picture after changing some colors can looks as follows:

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

In the second test sample the picture after changing some colors can looks as follows:

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