A

Plants vs Zombies

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

Prof. Pang is playing *Plants vs Zombies*.

Imagine that the game is played on a number axis. The following are the elements in the game:

- n zombies. The i-th zombie appears at 0 on the number axis at time t_i with health point h_i. The zombies have the same moving speed V and they all move to the right.
- m spikeweeds. The i-th spikeweed is of position p_i and attack power a_i.
- One peashooter at the position of 10^{100}. It shoots K peas of attack power D every second.

Every second in the game is processed as follows:

1. When the x-th second begins, the zombies whose t_is equal x appear at position 0.
2. After that, for each appeared and alive zombie u, it will suffer from the spikeweeds whose positions are in $(P_u, P_u + V]$ where P_u is the current position of the u-th zombie. So its health point will be decreased by $\sum_{1 \le i \le m, P_u < p_i \le P_u + V} a_i$. The zombie dies if its health point is no more than zero. Otherwise, it is still alive and its position will be increased by V.
3. When the x-th second ends, the peashooter shoots K peas in a row. For each pea, it will hit the zombie that is alive and of the maximum position currently. If there are multiple zombies of the maximum position, the pea hits the one of the minimum index. The health point of the zombie being hit decreases by D. This zombie dies if its health point is decreased to some value no more than 0. The peas are processed one by one, not simultaneously. (For example, if a zombie is killed by the first pea, the second pea cannot hit it since it dies before the second pea is shot.) If no alive zombie exists, the remain peas will hit nothing.

Prof. Pang wants to know the death time (in seconds) of all the n zombies.

Input

The first line contains five integers n, m, V, K, D ($1 \le n, m \le 10^5, 1 \le V, K, D \le 10^9$) separated by single spaces.

Each of the following n lines contains two integers t_i, h_i ($1 \le t_i, h_i \le 10^9$) separated by a single space.

Each of the following m lines contains two integers p_i, a_i ($1 \le p_i, a_i \le 10^9$) separated by a single space.

Output

Output one line containing n integers, where the i-th integer denotes the death time (in seconds) of the i-th zombie.

Example

input
3 2 1 2 2 1 11 2 8 1 1 1 2 2 4
output
2 3 1

Note

During the first second:

- The first zombie appears and then moves to position 1. It suffers 6 damage points (2 from the first spikeweed, 4 from the two peas).

- The third zombie appears and then moves to position 1. It suffers 2 damage points (from the first spikeweed) and dies (since its health point becomes -1).

During the second second:

- The first zombie moves to position 2 and suffers 6 damage points (4 from the second spikeweed, 2 from the first pea) and dies (since its health point becomes -1).
- The second zombie appears and then moves to position 1. It suffers 4 damage points (2 from the first spikeweed, 2 from the second pea).

During the third second:

- The second zombie moves to position 2, suffers 4 damage points (4 from the second spikeweed) and dies (since its health point becomes 0).
- The peas hit no zombie during this second.

So the death times of the zombies are 2, 3, 1, respectively.

B DZY Loves Chemistry

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

DZY loves chemistry, and he enjoys mixing chemicals.

DZY has n chemicals, and m pairs of them will react. He wants to pour these chemicals into a test tube, and he needs to pour them in one by one, in any order.

Let's consider the danger of a test tube. Danger of an empty test tube is 1. And every time when DZY pours a chemical, if there are already one or more chemicals in the test tube that can react with it, the danger of the test tube will be multiplied by 2. Otherwise the danger remains as it is.

Find the maximum possible danger after pouring all the chemicals one by one in optimal order.

Input

The first line contains two space-separated integers n and m $(1 \le n \le 50;\ 0 \le m \le \frac{n(n-1)}{2})$.

Each of the next m lines contains two space-separated integers x_i and y_i $(1 \le x_i < y_i \le n)$. These integers mean that the chemical x_i will react with the chemical y_i. Each pair of chemicals will appear at most once in the input.

Consider all the chemicals numbered from 1 to n in some order.

Output

Print a single integer — the maximum possible danger.

Examples

input
1 0
output
1

input
2 1 1 2
output
2

input
3 2 1 2 2 3
output
4

Note

In the first sample, there's only one way to pour, and the danger won't increase.

In the second sample, no matter we pour the 1st chemical first, or pour the 2nd chemical first, the answer is always 2.

In the third sample, there are four ways to achieve the maximum possible danger: 2-1-3, 2-3-1, 1-2-3 and 3-2-1 (that is the numbers of the chemicals in order of pouring).

C Definite Game

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

Chouti was doing a competitive programming competition. However, after having all the problems accepted, he got bored and decided to invent some small games.

He came up with the following game. The player has a positive integer n. Initially the value of n equals to v and the player is able to do the following operation as many times as the player want (possibly zero): choose a positive integer x that $x < n$ and x is not a divisor of n, then subtract x from n. The goal of the player is to minimize the value of n in the end.

Soon, Chouti found the game trivial. Can you also beat the game?

Input

The input contains only one integer in the first line: v ($1 \leq v \leq 10^9$), the initial value of n.

Output

Output a single integer, the minimum value of n the player can get.

Examples

input
8
output
1

input
1
output
1

Note

In the first example, the player can choose $x = 3$ in the first turn, then n becomes 5. He can then choose $x = 4$ in the second turn to get $n = 1$ as the result. There are other ways to get this minimum. However, for example, he cannot choose $x = 2$ in the first turn because 2 is a divisor of 8.

In the second example, since $n = 1$ initially, the player can do nothing.

D Word Cut

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

Let's consider one interesting word game. In this game you should transform one word into another through special operations.

Let's say we have word w, let's split this word into two non-empty parts x and y so, that $w = xy$. A *split* operation is transforming word $w = xy$ into word $u = yx$. For example, a *split* operation can transform word "wordcut" into word "cutword".

You are given two words $start$ and end. Count in how many ways we can transform word $start$ into word end, if we apply exactly k *split* operations consecutively to word $start$.

Two ways are considered different if the sequences of applied operations differ. Two operation sequences are different if exists such number i ($1 \le i \le k$), that in the i-th operation of the first sequence the word splits into parts x and y, in the i-th operation of the second sequence the word splits into parts a and b, and additionally $x \neq a$ holds.

Input

The first line contains a non-empty word $start$, the second line contains a non-empty word end. The words consist of lowercase Latin letters. The number of letters in word $start$ equals the number of letters in word end and is at least 2 and doesn't exceed 1000 letters.

The third line contains integer k ($0 \le k \le 10^5$) — the required number of operations.

Output

Print a single number — the answer to the problem. As this number can be rather large, print it modulo 1000000007 ($10^9 + 7$).

Examples

input
ab ab 2
output
1

input
ababab ababab 1
output
2

input
ab ba 2
output
0

Note

The sought way in the first sample is:

ab → a|b → ba → b|a → ab

In the second sample the two sought ways are:

- ababab → abab|ab → ababab

- `ababab` → `ab|abab` → `ababab`

E

Minimal Matrix

Time limit:	3 seconds
Memory limit:	256 megabytes

Research Institute of Given Strings is opening a new department — Research Deparment of Given Matrices. Similarly to a problem of string canonization the new deparment is now working on a problem of matrix canonization.

Consider a matrix $m_{i,j}$ of size $2^n \times 2^n$ containing lowercase letters.

The circular shift of a matrix m is a matrix m' such that $m'_{i,j} = m_{(i+\Delta i) \bmod 2^n, (j+\Delta j) \bmod 2^n}$ from some Δi and Δj (matrices are indexed from 0 to $2^n - 1$).

Matrix p is lexicographically less than matrix q of the same size if there are some i and j such that for $i' < i$, or $i' = i$ and $j' < j$ the equality $p_{i',j'} = q_{i',j'}$ holds, and $p_{i,j} < q_{i,j}$. That is, we compare matrices by rows.

Given matrix m, the problem of its canonization is to find its circular shift such that it is lexicographically less or equal to any other circular shift of m.

Help the researchers in the new department to find the canonization of a given matrix.

Input

The input file contains a matrix m. Its size if $2^n \times 2^n$, $0 \le n \le 9$.

Output

Output the canonization of the given matrix.

Example

matrix.in	matrix.out
baba baab abba bbba	aabb abbb abab bbaa

F Grandma Laura and Apples

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

Grandma Laura came to the market to sell some apples. During the day she sold all the apples she had. But grandma is old, so she forgot how many apples she had brought to the market.

She precisely remembers she had n buyers and each of them bought exactly half of the apples she had at the moment of the purchase and also she gave a half of an apple to some of them as a gift (if the number of apples at the moment of purchase was odd), until she sold all the apples she had.

So each buyer took some integral positive number of apples, but maybe he didn't pay for a half of an apple (if the number of apples at the moment of the purchase was odd).

For each buyer grandma remembers if she gave a half of an apple as a gift or not. The cost of an apple is p (the number p is even).

Print the total money grandma should have at the end of the day to check if some buyers cheated her.

Input

The first line contains two integers n and p ($1 \le n \le 40, 2 \le p \le 1000$) — the number of the buyers and the cost of one apple. It is guaranteed that the number p is even.

The next n lines contains the description of buyers. Each buyer is described with the string `half` if he simply bought half of the apples and with the string `halfplus` if grandma also gave him a half of an apple as a gift.

It is guaranteed that grandma has at least one apple at the start of the day and she has no apples at the end of the day.

Output

Print the only integer a — the total money grandma should have at the end of the day.

Note that the answer can be too large, so you should use 64-bit integer type to store it. In `C++` you can use the `long long` integer type and in `Java` you can use `long` integer type.

Examples

input
2 10 half halfplus
output
15

input
3 10 halfplus halfplus halfplus
output
55

Note

In the first sample at the start of the day the grandma had two apples. First she sold one apple and then she sold a half of the second apple and gave a half of the second apple as a present to the second buyer.

G Triangle

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

Johnny has a younger sister Anne, who is very clever and smart. As she came home from the kindergarten, she told his brother about the task that her kindergartener asked her to solve. The task was just to construct a triangle out of four sticks of different colours. Naturally, one of the sticks is extra. It is not allowed to break the sticks or use their partial length. Anne has perfectly solved this task, now she is asking Johnny to do the same.

The boy answered that he would cope with it without any difficulty. However, after a while he found out that different tricky things can occur. It can happen that it is impossible to construct a triangle of a positive area, but it is possible to construct a degenerate triangle. It can be so, that it is impossible to construct a degenerate triangle even. As Johnny is very lazy, he does not want to consider such a big amount of cases, he asks you to help him.

Input

The first line of the input contains four space-separated positive integer numbers not exceeding 100 — lengthes of the sticks.

Output

Output `TRIANGLE` if it is possible to construct a non-degenerate triangle. Output `SEGMENT` if the first case cannot take place and it is possible to construct a degenerate triangle. Output `IMPOSSIBLE` if it is impossible to construct any triangle. Remember that you are to use three sticks. It is not allowed to break the sticks or use their partial length.

Examples

input
4 2 1 3
output
TRIANGLE

input
7 2 2 4
output
SEGMENT

input
3 5 9 1
output
IMPOSSIBLE

H Divisors

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

You are given n integers $a_1, a_2, \ldots, a_n$. Each of a_i has between 3 and 5 divisors. Consider $a = \prod a_i$ — the product of all input integers. Find the number of divisors of a. As this number may be very large, print it modulo prime number 998244353.

Input

The first line contains a single integer n ($1 \le n \le 500$) — the number of numbers.

Each of the next n lines contains an integer a_i ($1 \le a_i \le 2 \cdot 10^{18}$). It is guaranteed that the number of divisors of each a_i is between 3 and 5.

Output

Print a single integer d — the number of divisors of the product $a_1 \cdot a_2 \cdot \cdots \cdot a_n$ modulo 998244353.

Hacks input

For hacks, the input needs to be provided in a special format.

The first line contains an integer n ($1 \le n \le 500$) — the number of numbers.

Each of the next n lines contains a prime factorization of a_i. The line contains an integer k_i ($2 \le k_i \le 4$) — the number of prime factors of a_i and k_i integers $p_{i,j}$ ($2 \le p_{i,j} \le 2 \cdot 10^{18}$) where $p_{i,j}$ is the j-th prime factor of a_i.

Before supplying the input to the contestant, $a_i = \prod p_{i,j}$ are calculated. Note that each $p_{i,j}$ must be prime, each computed a_i must satisfy $a_i \le 2 \cdot 10^{18}$ and must have between 3 and 5 divisors. The contestant will be given only a_i, and not its prime factorization.

For example, you need to use this test to get the first sample:

```
3
2 3 3
2 3 5
2 11 13
```

Interaction

From the technical side, this problem is interactive. Therefore, do not forget to output end of line and flush the output. Also, do not read more than you need. To flush the output, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

Examples

input

```
3
9
15
143
```

output

```
32
```

input

```
1
7400840699802997
```

output

```
4
```

input

```
8
4606061759128693
4606066102679989
4606069767552943
4606063116488033
4606063930903637
4606064745319241
4606063930904021
4606065559735517
```

output

```
1920
```

input

```
3
4
8
16
```

output

```
10
```

Note

In the first case, $a = 19305$. Its divisors are $1, 3, 5, 9, 11, 13, 15, 27, 33, 39, 45, 55, 65, 99, 117, 135, 143, 165, 195, 297, 351, 429, 495, 585, 715, 1287, 1485, 1755, 2145, 3861, 6435, 19305$ — a total of 32.

In the second case, a has four divisors: 1, 86028121, 86028157, and 7400840699802997.

In the third case $a = 2026004456719253646987390616290838779819620697031402685165705648886993752094772140451022537660230724015574910544536902134835473$.

In the fourth case, $a = 512 = 2^9$, so answer equals to 10.

I

Maximum Subsequence

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

You are given an array a consisting of n integers, and additionally an integer m. You have to choose some sequence of indices $b_1, b_2, ..., b_k$ ($1 \le b_1 < b_2 < ... < b_k \le n$) in such a way that the value of $\sum_{i=1}^{k} a_{b_i} \bmod m$ is maximized. Chosen sequence can be empty.

Print the maximum possible value of $\sum_{i=1}^{k} a_{b_i} \bmod m$.

Input

The first line contains two integers n and m ($1 \le n \le 35$, $1 \le m \le 10^9$).

The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

Output

Print the maximum possible value of $\sum_{i=1}^{k} a_{b_i} \bmod m$.

Examples

input
4 4 5 2 4 1
output
3

input
3 20 199 41 299
output
19

Note

In the first example you can choose a sequence $b = \{1, 2\}$, so the sum $\sum_{i=1}^{k} a_{b_i}$ is equal to 7 (and that's 3 after taking it modulo 4).

In the second example you can choose a sequence $b = \{3\}$.

J

Toll Roads

time limit per test: 10 seconds
memory limit per test: 256 mebibytes
input: *standard input*
output: *standard output*

There are n cities in the country of Flatland, n - 1 bidirectional roads connect some pairs of Flatland's cities. It is possible to get from any city to any other city of the country using roads. In this problem we will refer to the smallest set of roads needed to travel from a to b as a *simple path*.

The government of Flatland has recently introduced payment tolls on all roads. Using one road connecting two cities costs one ruble. That means that every time when you travel from one city to another one, you have to pay as many rubles as the number of roads you pass during your journey.

Many people got upset because of this decision, especially people who travel long distances. Political opponents of the current government claim that the maximal cost of a simple path in Flatland is very high, namely $cost$ rubles. The government has decided to support people and calm down the opposition. They want to reduce the maximal cost of a simple path in the country as much as possible. In other words, they want the number $cost$ reported by the opposition to be the lowest possible. The government will allow to remove tolls from at most k roads in order to achieve that. If there are multiple ways to do that, they want to minimize the number of roads that will become free.

Like any other government, Flatland's one is quite inflexible. They additionally require that it should be possible to represent the set of roads which will become free as a simple path between some cities x and y. Your task is to help the government.

Input

The first line contains two integers n and k ($1 \le k < n \le 5000$), the number of cities in Flatland and the maximum number of roads to become toll-free, respectively. Each of the next n - 1 lines contains two integers u_i and v_i, meaning that there is a road connecting cities u_i and v_i ($0 \le u_i, v_i < n$).

Output

On the first line of the output, print one integer: the minimum $cost$ of the most expensive simple path in Flatland that the government can achieve ($0 \le cost < n - 1$). On the second line, print the minimum number of roads t that must be made free to achieve that $cost$ ($0 \le t \le k$). If $t > 0$, print two space-separated integers on the third line: the numbers of cities x and y such that all roads on the simple path between them must be made free ($0 \le x, y < n$, the simple path between x and y must contain exactly t roads).

Examples

input
8 3 0 2 0 5 2 3 5 1 4 5 5 6 6 7
output
2 3 2 6

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

K

Chores

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

Petya and Vasya are brothers. Today is a special day for them as their parents left them home alone and commissioned them to do n chores. Each chore is characterized by a single parameter — its complexity. The complexity of the i-th chore equals h_i.

As Petya is older, he wants to take the chores with complexity larger than some value x ($h_i > x$) to leave to Vasya the chores with complexity less than or equal to x ($h_i \le x$). The brothers have already decided that Petya will do exactly a chores and Vasya will do exactly b chores ($a + b = n$).

In how many ways can they choose an integer x so that Petya got exactly a chores and Vasya got exactly b chores?

Input

The first input line contains three integers n, a and b ($2 \le n \le 2000$; $a, b \ge 1$; $a + b = n$) — the total number of chores, the number of Petya's chores and the number of Vasya's chores.

The next line contains a sequence of integers $h_1, h_2, ..., h_n$ ($1 \le h_i \le 10^9$), h_i is the complexity of the i-th chore. The numbers in the given sequence are not necessarily different.

All numbers on the lines are separated by single spaces.

Output

Print the required number of ways to choose an integer value of x. If there are no such ways, print `0`.

Examples

input
5 2 3 6 2 3 100 1

output
3

input
7 3 4 1 1 9 1 1 1 1

output
0

Note

In the first sample the possible values of x are 3, 4 or 5.

In the second sample it is impossible to find such x, that Petya got 3 chores and Vasya got 4.

Trading business

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

To get money for a new aeonic blaster, ranger Qwerty decided to engage in trade for a while. He wants to buy some number of items (or probably not to buy anything at all) on one of the planets, and then sell the bought items on another planet. Note that this operation is not repeated, that is, the buying and the selling are made only once. To carry out his plan, Qwerty is going to take a bank loan that covers all expenses and to return the loaned money at the end of the operation (the money is returned without the interest). At the same time, Querty wants to get as much profit as possible.

The system has n planets in total. On each of them Qwerty can buy or sell items of m types (such as food, medicine, weapons, alcohol, and so on). For each planet i and each type of items j Qwerty knows the following:

- a_{ij} — the cost of buying an item;
- b_{ij} — the cost of selling an item;
- c_{ij} — the number of remaining items.

It is not allowed to buy more than c_{ij} items of type j on planet i, but it is allowed to sell any number of items of any kind.

Knowing that the hold of Qwerty's ship has room for no more than k items, determine the maximum profit which Qwerty can get.

Input

The first line contains three space-separated integers n, m and k ($2 \le n \le 10$, $1 \le m, k \le 100$) — the number of planets, the number of question types and the capacity of Qwerty's ship hold, correspondingly.

Then follow n blocks describing each planet.

The first line of the i-th block has the planet's name as a string with length from 1 to 10 Latin letters. The first letter of the name is uppercase, the rest are lowercase. Then in the i-th block follow m lines, the j-th of them contains three integers a_{ij}, b_{ij} and c_{ij} ($1 \le b_{ij} < a_{ij} \le 1000$, $0 \le c_{ij} \le 100$) — the numbers that describe money operations with the j-th item on the i-th planet. The numbers in the lines are separated by spaces.

It is guaranteed that the names of all planets are different.

Output

Print a single number — the maximum profit Qwerty can get.

Examples

input
3 3 10 Venus 6 5 3 7 6 5 8 6 10 Earth 10 9 0 8 6 4 10 9 3 Mars 4 3 0 8 4 12 7 2 5
output
16

Note

In the first test case you should fly to planet `Venus`, take a loan on 74 units of money and buy three

items of the first type and 7 items of the third type ($3 \cdot 6 + 7 \cdot 8 = 74$). Then the ranger should fly to planet `Earth` and sell there all the items he has bought. He gets $3 \cdot 9 + 7 \cdot 9 = 90$ units of money for the items, he should give 74 of them for the loan. The resulting profit equals 16 units of money. We cannot get more profit in this case.

M Colorful Field

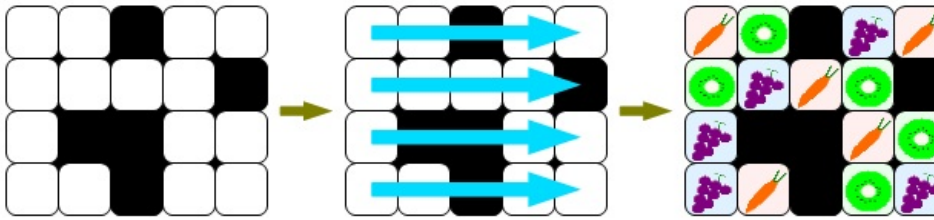
time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel saw a large field while she was on a bus. The field was a $n \times m$ rectangle divided into 1×1 cells. Some cells were wasteland, and other each cell contained crop plants: either carrots or kiwis or grapes.

After seeing the field carefully, Ciel found that the crop plants of each cell were planted in following procedure:

- Assume that the rows are numbered 1 to n from top to bottom and the columns are numbered 1 to m from left to right, and a cell in row i and column j is represented as (i, j).
- First, each field is either cultivated or waste. Crop plants will be planted in the cultivated cells in the order of $(1, 1) \rightarrow ... \rightarrow (1, m) \rightarrow (2, 1) \rightarrow ... \rightarrow (2, m) \rightarrow ... \rightarrow (n, 1) \rightarrow ... \rightarrow (n, m)$. Waste cells will be ignored.
- Crop plants (either carrots or kiwis or grapes) will be planted in each cell one after another cyclically. Carrots will be planted in the first cell, then kiwis in the second one, grapes in the third one, carrots in the forth one, kiwis in the fifth one, and so on.

The following figure will show you the example of this procedure. Here, a white square represents a cultivated cell, and a black square represents a waste cell.



Now she is wondering how to determine the crop plants in some certain cells.

Input

In the first line there are four positive integers n, m, k, t ($1 \le n \le 4 \cdot 10^4$, $1 \le m \le 4 \cdot 10^4$, $1 \le k \le 10^3$, $1 \le t \le 10^3$), each of which represents the height of the field, the width of the field, the number of waste cells and the number of queries that ask the kind of crop plants in a certain cell.

Following each k lines contains two integers a, b ($1 \le a \le n$, $1 \le b \le m$), which denotes a cell (a, b) is waste. It is guaranteed that the same cell will not appear twice in this section.

Following each t lines contains two integers i, j ($1 \le i \le n$, $1 \le j \le m$), which is a query that asks you the kind of crop plants of a cell (i, j).

Output

For each query, if the cell is waste, print `Waste`. Otherwise, print the name of crop plants in the cell: either `Carrots` or `Kiwis` or `Grapes`.

Examples

input
4 5 5 6 4 3 1 3 3 3 2 5 3 2 1 3 1 4 2 3 2 4 1 1 1 1

output

```
Waste
Grapes
Carrots
Kiwis
Carrots
Carrots
```

Note

The sample corresponds to the figure in the statement.