Problem A. Scott's New Trick

Input file:	standard input
Output file:	standard output
Time limit:	30 seconds
Memory limit:	256 megabytes

Little Scott recently learned how to perform arithmetic operations modulo some prime number P. As a training set he picked two sequences a of length N and b of length M, generated in the following way:

- $a_1 = A_1$, $a_2 = A_2$, $a_i = (a_{i-2} \cdot A_3 + a_{i-1} \cdot A_4 + A_5) \bmod P$, for $i \in [3, N]$.
- $b_1 = B_1$, $b_2 = B_2$, $b_j = (b_{j-2} \cdot B_3 + b_{j-1} \cdot B_4 + B_5) \bmod P$, for $j \in [3, M]$.

Now he wants to find the number of pairs (i, j), where $1 \le i \le N$ and $1 \le j \le M$, such that $(a_i \cdot b_j) \bmod P < L$, for given number L.

He asked you to do the same to help him check his answers.

Input

The first line of the input contains two integers: prime number P and positive integer L ($2 \le P < 250\,000$, $1 \le L \le P$). The second line consists of six non-negative integers N, A_1, A_2, A_3, A_4, A_5. Likewise, the third line contains six non-negative integers M, B_1, B_2, B_3, B_4, B_5. ($1 \le N, M \le 10\,000\,000$, $0 \le A_1, A_2, A_3, A_4, A_5, B_1, B_2, B_3, B_4, B_5 \le P$)

Output

The sole line of the output should contain one integer — the answer for the problem.

Examples

standard input	standard output
3 1 4 0 2 2 2 2 2 1 2 1 0 0	6
3 1 5 2 0 0 1 1 5 1 1 2 0 0	10
3 3 5 0 0 1 2 2 3 2 1 1 1 1	15
5 1 5 2 0 4 0 4 3 2 1 2 4 4	3
5 4 2 2 1 3 1 4 5 1 0 2 3 3	9

Problem B. Cryptography Entertainment

Input file:	standard input
Output file:	standard output
Time limit:	5 seconds
Memory limit:	256 megabytes

Vasya and Petya decided to play spies. Vasya came up with the word of length n consisting only of lower-case english letters. He gave to Petya only the polynomial hash x. Your task is to calculate the total number of possible initial words modulo 998244353.

Polynomial hash of string $s = a_0 a_1 \dots a_{n-1}$ is equal to $ord(a_0) + ord(a_1) \cdot p^1 + \dots + ord(a_{n-1}) \cdot p^{n-1} \bmod m$, where $ord(c)$ is a function of a position of a symbol: ord('a')= 1, ord('b')= 2, ... ord('z')= 26.

Input

The sole line of the input contains four integers n, m, p and x ($1 \le n \le 10^6$, $2 \le m \le 10^4$, $1 \le p < m$, $0 \le x < m$).

Output

The sole line of the output should contain one integer — the answer to the problem.

Examples

standard input	standard output
1 10 8 7	2

Problem C. Equal sums

Input file:	standard input
Output file:	standard output
Time limit:	15 seconds
Memory limit:	512 megabytes

I have a set of positive integers S. Can you find two non-empty, distinct subsets with the same sum?

Note: A subset is a set that contains only elements from S, and two subsets are distinct if they do not have exactly the same elements.

Input

The first line of the input contains N ($N = 500$) following by distinct elements of set S. Each element of set does not exceed 10^{12}.

Output

If there are two different subsets of S that have the same sum, then output these subsets, one per line. Each line should contain the size of the subset followed the numbers in one subset, separated by spaces. If it is impossible, then you should output the string "Impossible"on a single line.

If there are multiple ways of choosing two subsets with the same sum, any choice is acceptable.

standard input	standard output
20 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	2 1 2 1 3
20 120 266 858 1243 1657 1771 2328 2490 2665 2894 3117 4210 4454 4943 5690 6170 7048 7125 9512 9600	3 3117 4210 4943 3 2328 2894 7048

Problem D. Runs

Input file:	standard input
Output file:	standard output
Time limit:	7 seconds
Memory limit:	512 megabytes

I have a string S consisting of lower-case alphabetic characters, 'a' - 'z'. Each maximal sequence of contiguous characters that are the same is called a "run". For example, "bookkeeper"has 7 runs. How many different permutations of S have exactly the same number of runs as S?

Two permutations a and b are considered different if there exists some index i at which they have a different character: $a[i] \neq b[i]$.

Input

The sole line of the input contains a single non-empty string of lower-case alphabetic characters, S, the string of interest. ($1 \leq |S| \leq 450\,000$). S has at most 100 runs.

Output

The sole line of the output should contain the answer for the problem modulo 1 000 003.

Examples

standard input	standard output
aabcd	24
bookkeeper	7200

Problem E. Number Game

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Arya and Bran are playing a game. Initially, two positive integers A and B are written on a blackboard. The players take turns, starting with Arya. On his or her turn, a player can replace A with $A - k \cdot B$ for any positive integer k, or replace B with $B - k \cdot A$ for any positive integer k. The first person to make one of the numbers drop to zero or below loses.

For example, if the numbers are initially $(12, 51)$, the game might progress as follows:

- Arya replaces 51 with $51 - 3 \cdot 12 = 15$, leaving $(12, 15)$ on the blackboard.
- Bran replaces 15 with $15 - 1 \cdot 12 = 3$, leaving $(12, 3)$ on the blackboard.
- Arya replaces 12 with $12 - 3 \cdot 3 = 3$, leaving $(3, 3)$ on the blackboard.
- Bran replaces one 3 with $3 - 1 \cdot 3 = 0$, and loses.

We will say (A, B) is a winning position if Arya can always win a game that starts with (A, B) on the blackboard, no matter what Bran does.

Given four integers A_1, A_2, B_1, B_2, count how many winning positions (A, B) there are with $A_1 \leq A \leq A_2$ and $B_1 \leq B \leq B_2$.

Input

The sole line of the input contains the four integers A_1, A_2, B_1, B_2, separated by spaces. ($1 \leq A_1 \leq A_2 \leq 10^6$, $1 \leq B_1 \leq B_2 \leq 10^6$, $A_2 - A_1 \leq 10^6 - 1$, $B_2 - B_1 \leq 10^6 - 1$).

Output

The sole line of the output should contain the answer for the problem.

Examples

standard input	standard output
5 5 8 8	0
11 11 2 2	1
1 6 1 6	20

Problem F. Your Rank is Pure

Input file:	standard input
Output file:	standard output
Time limit:	10 seconds
Memory limit:	64 megabytes

You are given a subset S of positive integers. A number in S is considered pure with respect to S if, starting from it, you can continue taking its rank in S, and get a number that is also in S, until in finite steps you hit the number 1, which is not in S.

When n is given, in how many ways you can pick S, a subset of $\{2, 3, ..., n\}$, so that n is pure, with respect to S? The answer might be a big number, you need to output it modulo 100 003.

Input

The first line of the input contains the number of test cases T ($1 \leq T \leq 100$). Each of the next T lines contains the description of the test case: a single integer n ($1 \leq n \leq 500$).

Output

The output should contain T lines. Each line should contain the answer to the corresponding test.

Example

standard input	standard output
2 5 6	5 8

Problem G. Different Sum

Input file:	standard input
Output file:	standard output
Time limit:	40 seconds
Memory limit:	256 megabytes

We define a cryptarithm equation to be an addition equation written in such a way that all summands (numbers being added) and the sum are aligned to the same right border like this:

```
 124
  31
  25
 ---
 180
```

Additionally, for each column of a cryptarithm equation, all digits of the summands in that column must be different. Note that we don't include the sum in this constraint. So for example in the above equation the first column contains only digit 1, the second column contains digits 2,3 and 2, and the third column contains digits 4, 1 and 5. This equation is not a cryptarithm equation since the second column contains two 2's. However, it would be a cryptarithm equation if we replaced the last summand with 15 (and the sum with 170).

Note that summands in a cryptarithm equation are always positive and written without leading zeros. The order of summands is not important (in other words, two equations which differ only in the order of the summands are considered the same).

The example above was in base 10, but we're also interested in cryptarithm equations in other bases. Note that a "digit"in base b could mean any integer between 0 and $b-1$. Here is a cryptarithm equation in base 23:

```
 I7B
 JJJ
```

```
----
1F47
```

In this example, "I"stands for digit 18, "B"stands for digit 11, "J"stands for digit 19, and "F"stands for digit 15. In decimal notation, the two summands are $18 \cdot 23^2 + 7 \cdot 23 + 11 = 9694$ and $19 \cdot 23^2 + 19 \cdot 23 + 19 = 10507$, and the sum $1 \cdot 23^3 + 15 \cdot 23^2 + 4 \cdot 23 + 7 = 20201$. Please note that denoting digits of 10 and more with letters was done purely for the clarity of the example; it doesn't really matter in this problem how exactly we denote such digits in writing.

How many cryptarithm equations are there with the given sum N in the given base B?

Since the answer might be very large, please output it modulo $10^9 + 7$.

Input

The sole line of the input contains N and B ($1 \le N \le 10^{18}$, $2 \le B \le 70$). All input numbers are given in base 10.

Output

The sole line of the output should contain the answer to the problem.

Examples

standard input	standard output
6 10	4
8 4	4

Note

Here are the 4 cryptarithm equations with sum 6:

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

And here are the 4 cryptarithm equations in base 4 with sum $8 = 20_4$:

```
20  11  13  10
--   3   1   3
20  --  --   1
    20  20  --
            20
```

Problem H. Doubly-sorted Grid

Input file:	standard input
Output file:	standard output
Time limit:	15 seconds
Memory limit:	512 megabytes

A rectangular grid with lower case English letters in each cell is called **doubly sorted** if in each row the letters are non-decreasing from the left to the right, and in each column the letters are non-decreasing from the top to the bottom. In the following examples, the first two grids are doubly sorted, while the other two are not:

```
abc    ace    aceg    base
def    ade    cdef    base
ghi    bdg    xxyy    base
```

You are given a partially-filled grid, where some of the cells are filled with letters. Your task is to compute the number of ways you can fill the rest of the cells so that the resulting grid is doubly sorted. The answer might be a big number; you need to output the number of ways modulo 10007.

Input

The first line of the input contains two integers R and C ($1 \le R, C \le 10$) — the number of rows and the number of columns, respectively. Next R lines contains the description of partially-filled grid. Each of the next R lines contains a string of length C. Each character of the string is either a lower-case English letter, or '.', indicating that the cell is not filled yet.

Output

The sole line of the output should contain the answer to the problem.

Examples

standard input	standard output
2 2 ad c.	23
3 3 .a. a.z .z.	7569
4 4 .... .g.. .cj.	0

Problem I. The Reverse Problem About The Turtle

Input file:	standard input
Output file:	standard output
Time limit:	3 seconds
Memory limit:	256 megabytes

One of the most known problem on dynamic programming is the problem about a turtle. Here we give its description.

You are given a field $n \times m$. For each cell you know is it forbidden or not. (Upper-left and bottom-down corners are always non-forbidden) The turtle appears at the upper-left corner. Each turn the turtle could make a move to its non-forbidden neighbour: one cell right or one cell down. You have to calculate the number of ways the turtle could get from the upper-left corner to the bottom-down corner.

You have to solve the reverse problem. You have to find a field for which the number of ways is exactly k.

Input

The sole line of the input contains k ($1 \le k \le 10^{18}$).

Output

The first line of the output should contain two integers n and m. They have to be greater than zero and not bigger than 300.

The next n lines should contain m symbols each — the description of the field. If the cell is non-forbidden it should contain '1', otherwise, '0'. Upper-left and bottom-down corners should be non-forbidden.

If there exist multiple fields, you could output any of them. There always exists at least one field.

Examples

standard input	standard output
2	2 3 111 011