Agha. KingdomTour

The king loves his queen a lot, and he wants to take her all over the kingdom, in just a single day.

The kingdom has N cities in it. There are bi-directional roads connecting the cities such that there is a unique path from any city in the kingdom to any other city. Each road has an integral length.

The king wants to take his queen on a tour in the kingdom which starts and ends at the kingdom's capital. Since the kingdom is really beautiful, the king wants the tour to cover each road in the kingdom at least once (it doesn't matter in which direction). However, he has been informed that this might take a really long time. Hence, he has decided to build at most K shortcuts. A shortcut can connect any two cities (even if they are directly connected by a road), and the length of a shortcut is L. Since the scenery around the newly constructed shortcuts might not be so beautiful, the king has demanded that any particular shortcut cannot be taken more than once during the tour.

The cities in the kingdom are numbered 0 to $N-1$. City 0 is always the capital. For each edge, there are three integers a, b, c indicating that there is a road of length c connecting city a and city b.

Find out the length of the shortest tour satisfying the king's demands.

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

There are four integers N, M, K, L ($2 \leq N \leq 200, 0 \leq K \leq 100, 1 \leq L \leq 10000$). Next M lines contains three integers each, say a_i, b_i, c_i, where $0 \leq a_i, b_i \leq N-1$, and $1 \leq c_i \leq 10000$. The roads will be such that the kingdom will satisfy the properties mentioned in the problem statement.

Output

For each test case, output the desired number in a line.

Sample Input

```
3 2 8 4
2 1 9
0 1 3
2 1 2 3
0 1 4
6 5 2 5
4 0 4
2 0 4
2 5 4
4 3 10
1 2 10
10 9 2 4
1 2 2
4 1 9
2 5 5
6 5 4
1 7 7
7 3 1
2 0 2
5 8 5
9 5 6
```

Sample Output

```
16
7
41
59
```

Buri. PrefixFreeCode

A prefix-free code of size N which uses K characters is a set of N distinct strings such that

- each string of the set contains only characters '0','1', ..., ('0'+$K-1$).
- no string of the set is a prefix of any other string of the set.

The cost of a prefix-free code can be calculated as sum of costs of characters used to write down all strings of the set.

You are given the size of the code N and the costs of using the characters `characterCosts`. Return the minimal possible cost of a prefix-free code of size N which uses these character costs.

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

There are two integers N, K ($2 \leq N \leq 500, 2 \leq K \leq 50$). Next K integers $c_1, c_2, \ldots, c_K$ denotes the cost of each characters.

Output

For each test case, please output the desired answer in a line.

Sample Input

```
3 2
1 4
3 3
1 3 5
500 50
2 4 6 8 10 12 14 16 18 20
22 24 26 28 30 32 34 36 38 40
42 44 46 48 50 52 54 56 58 60
62 64 66 68 70 72 74 76 78 80
82 84 86 88 90 92 94 96 98 100
500 10
1 2 3 4 5 6 7 8 9 10
```

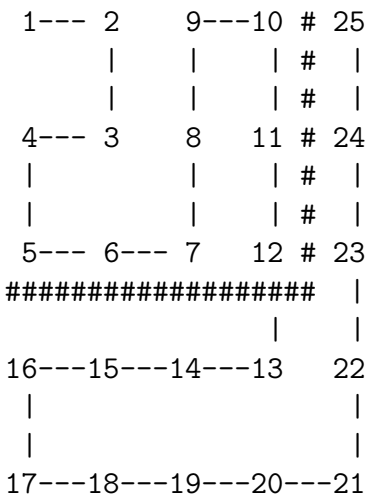
Sample Output

```
11
9
9464
4732
```

(Almost empty page.)

Clay. MegaSum

Let us define an infinite table of integers as follows:



Each cell is uniquely identified by the value it contains. Let us define $S(X)$ as the sum of all the values in the rectangle with cell 1 as its upper-left corner and cell X as its lower-right corner. For example, $S(12)$ is the sum of all the values in the sharp-ed area shown above.

You are given a long long integer N. First, find the rectangle with cell 1 as its upper-left corner and cell N as its lower-right corner. Then, calculate the sum of $S(X)$ for all values X inside this rectangle. Return this sum modulo $1,000,000,007$.

For example, if N is 8, you would first find the 3×2 rectangle with 1 in its upper-left corner and 8 in its lower-right corner. You would then calculate $S(X)$ for each value X in this rectangle: $S(1) = 1$, $S(2) = 3$, $S(9) = 12$, $S(4) = 5$, $S(3) = 10$, and $S(8) = 27$. You would them sum these values to get $1 + 3 + 12 + 5 + 10 + 27 = 58$.

Input

There are multiple test cases in the input file terminated by EOF. For each test case, there is an integer N in a line ($1 \leq N \leq 10^{10}$).

Output

For each test case, please output the desired number.

Sample Input	Sample Output
8	58
12	282
11	128
6	50
34539	437909839

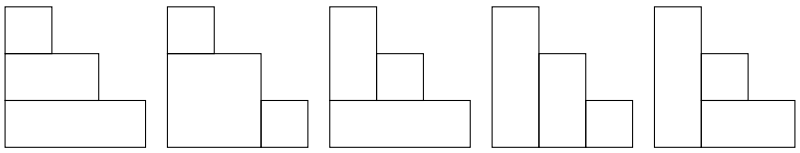
(Almost empty page.)

Dust. StairsColoring

A very rich sultan built an enormous luxurious two-story palace containing several staircases. According to an old tradition, each staircase must:

- contain exactly N steps
- have a right angle in its base
- be built using exactly N rectangular blocks of arbitrary size

Staircases can be built using many different arrangements of blocks. For example, there are 5 ways to build a staircase with 3 steps:



To ensure that his palace is really the most luxurious in the world, the sultan decided to build one staircase for each possible arrangement of blocks.

The sultan is now preparing for a staircase coloring festival. He wants to paint each of the staircases in the palace in one of K different colors. Multiple staircases can be painted the same color, and it is not necessary to use all K colors. Help the sultan by calculating the total number of different ways he can color his staircases. The answer can be large, so return the count modulo 1000000123.

Input

There are multiple test cases in the input file terminated by EOF. For each test case, there are two integers N, K in a line. $(1 \leq N, K \leq 10^9)$.

Output

For each test case, please output the desired number.

Sample Input

```
3 2
2 2
1 1
4 5
7 77
```

Sample Output

```
32
4
1
103514887
747707397
```

(Almost empty page.)

Ethe. CellScores

You are given a square board of size $N \times N$, where each 1×1 cell is either black or white. A rectangle on the board is called good if it contains only white cells. The score of each cell is equal to the number of distinct good rectangles that contain that cell.

Use the following instructions to determine the coloring of the board. You are given ints M, K, $X0$, $Y0$, A, B, C and D. Generate two lists, X and Y, each of length $M + K$, using the following recursive definitions:

- $X_0 = X0 \bmod N$
- $X_i = (X_{i-1} \times A + B) \bmod N$
- $Y_0 = Y0 \bmod N$
- $Y_i = (Y_{i-1} \times C + D) \bmod N$

For each i between 0 and $M - 1$, inclusive, the cell at column X_i, row Y_i is black. The remaining cells are white.

Let's denote the score for the cell at column i, row j as score(i, j). Return the sum of score(X_i, Y_i) for all i between M and $M + K - 1$, inclusive.

Input

There are multiple test cases in the input file terminated by EOF. For each test case, there are 9 integers $N, M, K, X0, A, B, Y0, C, D$ in a line. $(1 \le N \le 1500; 0 \le M, K \le 1000000; 0 \le X0, Y0, A, B, C, D \le 1500)$

Output

For each test case, please output the desired number. Note that the answer will be less than 2^{63}.

Sample Input

```
10 0 1 0 1 1 0 1 1
10 1 1 5 1 5 5 1 5
7 4 3 0 1 1 0 1 1
23 10 30 26 48 76 231 463 707
211 30 12 3 35 82 0 43 15
3 0 100 0 0 0 0 0 0
```

Sample Output

```
100
75
194
8088
18196443
900
```

(Almost empty page.)

Forc. TheEncryptionDivOne

John is obsessed with security. He is writing a letter to his friend Brus and he wants nobody else to be able to read it. He uses a simple substitution cipher to encode his message. Each letter in the message is replaced with its corresponding letter in a substitution alphabet. A substitution alphabet is a permutation of all the letters in the original alphabet. In this problem, the alphabet will consist of only lowercase and uppercase letters ('a'–'z', 'A'–'Z').

John wants to be sure that his encryption is safe, so he will not choose a cipher where a letter is encoded to either itself or to its lowercase or uppercase equivalent. For example, he will not choose a cipher where the letter 'j' is encoded to either 'j' or 'J'.

Given the original message msg and encoded message $encMsg$, determine the number of simple substitution ciphers that fit John's requirements and encode msg to $encMsg$. Return this number modulo 1234567891.

Input

There are multiple test cases in the input file terminated by EOF. For each test case, there are two strings, one contains msg and another one contains $encMsg$. They have same length and the length is between 1 and 50 inclusively. Both strings will contain lower- or upper-case letters only.

Output

For each test case, please output the desired number.

Sample Input

```
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWX
cdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWX
bcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXY
topcoder
TOPCODER
thisisaveryhardproblem
nobodywillsolveittoday
```

Sample Output

```
2
1
0
0
```

(Almost empty page.)

Gaun. AvoidFour

It is a known fact that of all numbers, 4 is the one that brings the worst luck. It is for this reason that when generating number sequences, we need to avoid patterns related to the number 4 as much as possible.

You are given a long long integer n. Count the number of positive integers that satisfy all of the following conditions:

- The number contains at most n digits.
- The number does not contain four consecutive '4' digits. For example, 43444124 is allowed, but 45444474 is not.
- The number of digits in the number is not a multiple of any of the integers greater than 10 that contain only '4' in their decimal representations (44, 444, 4444, 44444, ...).

Return the total count of these numbers modulo 1000000007.

Input

There are multiple test cases in the input file terminated by EOF. For each test case, there is a number n in a line. ($1 \leq n \leq 4 \times 10^{10}$).

Output

For each test case, please output the desired number.

Sample Input	Sample Output
4	9998
5	99980
87	576334228
88	576334228

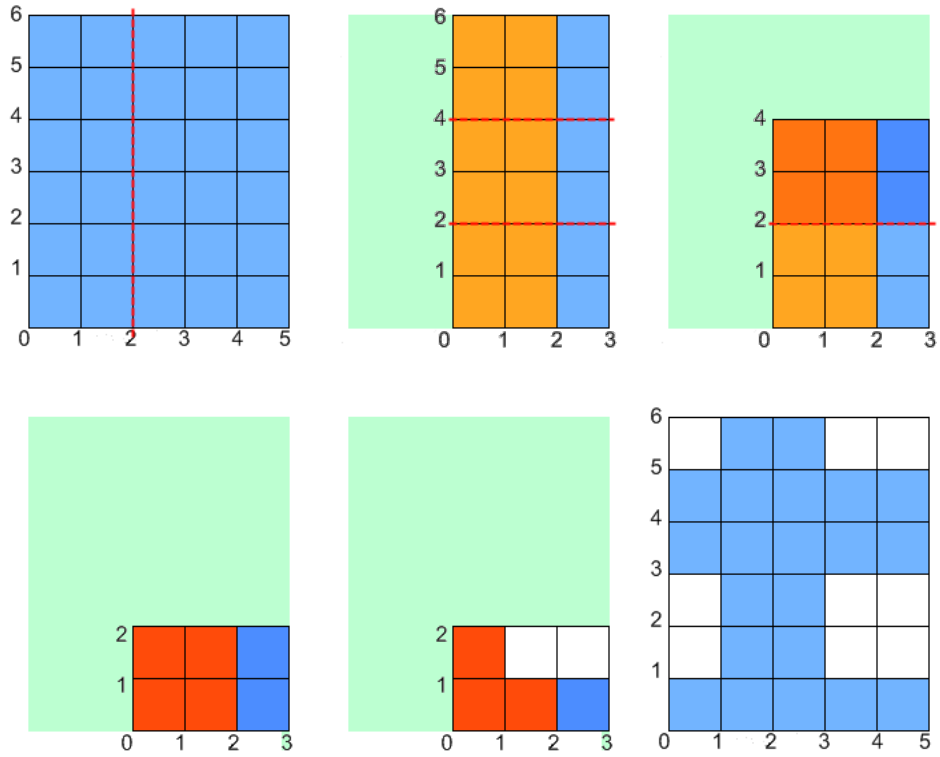
(Almost empty page.)

Hood. PaperAndPaint

Onise likes to play with paper and paint. He has a piece of paper with dimensions *width* × *height*. He performs K operations, one for each i between 0 and $K-1$, inclusive. Each operation consists of the following steps:

1. Fold the paper along the line $x =$ `xfold[`i`]` (the left side of the paper is folded over the right side).
2. Divide the paper vertically into `cnt[`i`]` +1 equal sections. Then, `cnt[`i`]` times, take the topmost section and fold it over the section below it.
3. Paint a rectangle with the lower-left corner at (`x1[`i`]`, `y1[`i`]`) and the upper-right corner at (`x2[`i`]`, `y2[`i`]`). Note that $(0, 0)$ is now the lower-left corner of the paper in its current folded state, not its original state. The paint will seep through all the layers of the folded paper. See the image below for clarification.
4. Unfold the paper.

For example, let's say Onise has a piece of paper that is 5×6. He performs one operation where `xfold` is 2, `cnt` is 2, and the coordinates of the painted rectangle's corners are $(1, 1)$ and $(3, 2)$. The following will happen (note that the paper starts out blue in the images and gets painted white):



Input

There are multiple test cases in the input file terminated by EOF. For each test case:

First line will contain three integers *width*, *height* and n. ($1 \leq width, height \leq 10^9; 1 \leq n \leq 50$). For next 6 lines, each line contains n integers denoting arrays `xfold[]`, `cnt[]`, `x1[]`, `y1[]`, `x2[]`, `y2[]` respectively. Every element in `xfold[]` will be between 0 and *width* inclusively. Every element in `cnt[]` will be between 0 and 1000 inclusively and each `cnt[`i`]` $+1$ will be a divisor of *height*. For every i, $0 \leq$ `x1[`i`]` $<$ `x2[`i`]` $\leq \max($`xfold[`i`]`, *width*$-$ `xfold[`i`]`$)$ and $0 \leq$ `y1[`i`]` $<$ `y2[`i`]` $\leq$ *height*$/($`cnt[`i`]`$+1))$.

Output

For each test case, return total area of paper that is not covered in paint.

Sample Input

```
5 6 1
2
2
1
1
3
2
2 4 2
0 0
1 0
0 0
1 1
2 1
2 4
6 6 2
2 5
1 2
1 2
2 0
3 4
3 2
21 30 5
3 21 7 11 13
4 14 9 5 4
4 0 2 5 9
2 0 1 2 3
7 19 6 11 12
5 2 2 4 5
30 42 5
16 24 25 21 4
5 1 6 13 20
3 1 5 8 9
0 1 0 0 1
14 22 12 18 13
2 15 1 1 2
26 60 5
17 17 24 4 21
4 1 11 0 2
9 1 20 18 7
1 3 0 45 12
13 4 23 19 13
3 14 1 46 14
17 3 6
17 2 10 2 10 13
2 0 0 2 0 0
7 6 4 11 0 5
0 0 1 0 1 1
12 10 6 12 4 12
1 3 2 1 2 2
```

Sample Output

```
21
3
18
27
336
1319
20
```

Jave. TheBeautifulBoard

You have a $n \times n$ board and several checkers of different colors. Let c_i be the number of checkers you have of color i for every $1 \leq i \leq m$. You want to place all of the checkers on the board in such a way that no cell contains more than one checker.

Return the number of different possible placements modulo 1234567891. Two placements are equal if you can get one from the other by rotating the board. Note that checkers of the same color are indistinguishable. If you have more checkers than the number of cells on the board, there are no possible placements, so you should return 0.

Note that you are not allowed to flip the board.

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

First line contains two integers n, m ($1 \leq n \leq 100000; 1 \leq m \leq 50$). Next m integers represent $c_1, \ldots, c_m$. Each $1 \leq c_i \leq 100000$.

Output

For each test case, please output the desired number.

Sample Input

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

Sample Output

```
1
3
18
0
```