Problem A. Fourier Transform

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Petya has recently learned the definition of the *partial Fourier transform* for the set of complex numbers. Now he asks you to help him perform partial Fourier transforms.

The definition of the partial Fourier transform follows.

Let $n = 2^s$, where $s \geq 0$ is an integer. First define the involution rev_s for integers from 0 to $2^s - 1$ the following way: $\mathrm{rev}_0(0) = 0$, $\mathrm{rev}_{s+1}(2x) = \mathrm{rev}_s(x)$, $\mathrm{rev}_{s+1}(2x+1) = 2^s + \mathrm{rev}_s(x)$ for $0 \leq x < 2^s$.

Now fix the n-th primitive root of unity $\zeta = e^{2\pi i/n} = \cos\frac{2\pi}{n} + i\sin\frac{2\pi}{n}$, and for any t from 0 to s define *the t-th partial Fourier transform* $\mathrm{Four}_s^{(t)}(\mathbf{a})$ of the array $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ of n complex numbers as: $\mathrm{Four}_s^{(t)}(\mathbf{a}) = \mathbf{b}^{(t)} = (b_0^{(t)}, b_1^{(t)}, \ldots, b_{n-1}^{(t)})$, where

$$b^{(t)}_{2^{s-t}j+k} = \sum_{j'=0}^{2^t-1} \zeta^{2^{s-t}jj'} a_{2^{s-t}j'+\mathrm{rev}_{s-t}(k)} \quad \text{for } 0 \leq j < 2^t,\ 0 \leq k < 2^{s-t}$$

So $\mathrm{Four}_s^{(s)} = \mathrm{Four}_s$ is the usual discrete Fourier transform

$$b_j^{(s)} = \sum_{j'=0}^{n-1} \zeta^{jj'} a_{j'} \quad ,$$

and $\mathrm{Four}_s^{(0)}$ just rearranges the numbers in the initial array with respect to rev: $b_k^{(0)} = a_{\mathrm{rev}_s(k)}$.

Petya thinks that it is easy to find $\mathrm{Four}_s^{(t)}$ if he knows $\mathrm{Four}_s^{(t-1)}$, but he hasn't found the required formula yet.

Help him to find the t-th partial Fourier transform of the given array of 2^s complex numbers.

Input

The first line of input contains s and t ($0 \leq t \leq s \leq 16$). The following 2^s lines contain two floating point numbers each — real and imaginary parts of a_j.

Output

Output 2^s lines with two floating point numbers each, real and imaginary part of $b_j^{(t)}$.

Example

standard input	standard output
2 1 2.00 0.00 7.00 0.00 3.00 0.00 -3.00 0.00	5.00000000 0.00000000 4.00000000 0.00000000 -1.00000000 0.00000000 10.00000000 0.00000000

Problem B. Multiplication

Input file:	*standard input*
Output file:	*standard ouput*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given two numbers. Find their product.

Input

Two lines contain one integer each. Length of each integer is no more than 250 000 symbols.

Output

Output one integer — product of given integers.

Examples

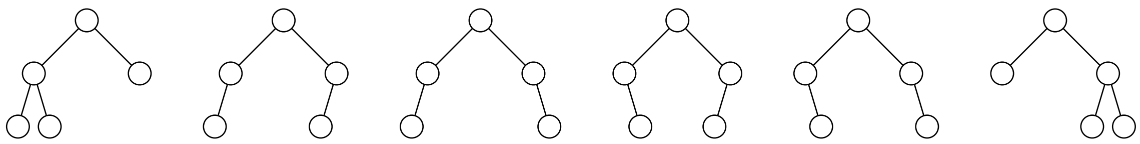
standard input	standard ouput
2 2	4

Problem C. AVL Trees

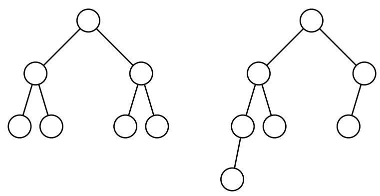
Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

AVL trees invented by Russian scientists Adelson-Velskiy and Landis are used for *sorted collection* data structure. The rooted binary tree is called *balanced* if for each vertex the height of its left subtree and the height of its right subtree differ by at most one. The balanced binary search tree is called the AVL tree.

There can be several AVL trees with the given number of vertices. For example, there are 6 AVL trees with 5 vertices, they are shown on the picture below.



Also the tree with the given number of vertices can have different height, the picture below shows AVL trees with 7 vertices that have height 2 and 3, respectively.



Given n and h find the number of AVL trees that have n vertices and height h. Since the answer can be quite large, return the answer modulo 786 433.

Input

Input file contains n and h ($1 \le n \le 65\,535$, $0 \le h \le 15$)

Output

Output one number — the number of AVL trees with n vertices that have height h, modulo 786 433.

Example

standard input	standard output
7 3	16

Note

Note that 786 433 is prime, and $786\,433 = 3 \cdot 2^{18} + 1$.

Problem D. Duel

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Alex and his rival George are preparing for the duel because of fair lady Nathalie. The duel will take place at a dark alley.

The alley has n trees and bushes growing along, the distance between adjacent plants is one meter. Alex and George decided that the duel will proceed as follows. Some tree is selected as the starting point and marked accordingly. Two trees at equal distance from the starting tree are marked as shooting points. Alex and George will start at the starting tree and move in opposite directions. When they reach shooting trees they will turn around and shoot at each other.

Given the positions of the trees, help Alex and George find the starting tree and shooting trees. First the duelists would like to know the number of ways they can choose the trees.

Input

Input file contains a non-empty string of 0-s and 1-s that describes the alley, 0 stands for the bush (that is not suitable to be neither starting nor shooting point), 1 stands for the tree. The length of the string doesn't exceed 100 000.

Output

Output the number of ways the duelists can choose starting and shooting trees.

Examples

standard input	standard output
101010101	4
101001	0

In the first example the following configurations of the duel are possible (starting and shooting trees are marked as bold): **1**0**1**0**1**0101, 10**1**0**1**0**1**01, 1010**1**0**1**0**1**, and **1**010**1**010**1**.

Problem E. Polynomial Hash

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

You are given n, m, p and x. Find how many different words of length n composed of letters of the English alphabet have a polynomial hash with base p equal to x modulo m. Find this number modulo 998 244 353.

Polynomial hash of a string $s = a_0 a_1 \ldots a_{n-1}$ is equal to

$$(ord(a_0) \cdot p^0 + ord(a_1) \cdot p^1 + \ldots + ord(a_{n-1}) \cdot p^{n-1}) \bmod m,$$

where $ord(c)$ is the ordinary number of c in the English alphabet, so $ord(a) = 1$, $ord(b) = 2$, $ord(z) = 26$.

Input

The first line of 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

Output one integer — the number of words of length n composed of letters of the English alphabet that have a polynomial hash with base p equal to x modulo m. The answer must be printed modulo 998 244 353.

Examples

standard input	standard output
1 10 8 7	2

Problem F. Costly Graphs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Let's define the cost of a simple undirected graph as the sum of the costs of its nodes. The cost of a node is defined as D^K, where D is its degree.

You are given N and K. You need to find the sum of the costs of all possible simple undirected graphs with N nodes. As this number may be very large, output the sum modulo 1005060097.

Input

The first line contains the number of test cases T ($1 \le T \le 2 \cdot 10^5$). Each of the next T lines contains two integers N and K ($1 \le N \le 10^9, 1 \le K \le 2 \cdot 10^5$) separated by a space. Sum of all K in one input file does not exceed $2 \cdot 10^5$

Output

For each test case, output one line containing the sum of the costs of all possible simple undirected graphs with N nodes, modulo 1005060097.

Examples

standard input	standard output
5 1 1 2 3 3 2 6 5 20 20	0 2 36 67584000 956922563

Problem G. Robot

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 mebibytes

There is a robot in a cell (x_1, y_1) of an infinite grid. It moves exactly t times to an adjacent cell, and gets to a cell (x_2, y_2). Two cells are adjacent if they have a common side.

It is known that while moving the robot always had positive x and y coordinates. It is also known that it was the first time the robot came to (x_2, y_2) after the t-th move.

You have to count the number of possible paths the robot could take that satisfy the conditions above. The answer can be quite large, so output it modulo 998 224 353.

Input

The first line of input contains five integers x_1, y_1, x_2, y_2, t. The cells (x_1, y_1) and (x_2, y_2) are distinct. $1 \le x_1, y_1, x_2, y_2, t \le 250\,000$.

Output

Output one integer — the number of possible paths of the robot, modulo 998 224 353.

Examples

standard input	standard output
`1 1 2 2 2`	`2`
`1 1 2 2 4`	`8`
`1 1 2 2 15`	`0`

Problem H. BankCraft

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 mebibytes

Osya and his gang want to expropriate some money from a backstreet millionaire Koreyko. But there is a problem: Koreyko stores all his money in a bank. The bank uses public-key cryptography to authenticate its clients. Each client has his own public key which is a polynomial $P(x)$ over the field of integers modulo p, and a secret key which is a polynomial $Q(x)$ over the same field. The secret key is considered to be valid if there exists a polynomial $R(x)$ such that $P(x) \cdot Q(x) = 1 + x^m \cdot R(x)$ for a known integer m.

Osya knows the polynomial $P(x)$ and integers p (it is always equal to $7\,340\,033$) and m, but he doesn't know the secret key. He offers you a key to the house full of money for your help with finding the secret key. You can't reject such a generous proposition.

Input

The first line of input contains two integers: m and n ($1 \leq m, n \leq 10^5$). Here, n is the degree of $P(x)$. The second line contains $n+1$ integers a_i ($0 \leq a_i \leq p-1$) separated by spaces, which are the coefficients of $P(x)$. The i-th of them ($0 \leq i \leq n$) is the coefficient of x^i.

Output

If it is impossible to construct the required polynomial of degree less than m, output the message "`The ears of a dead donkey`" (without quotes). If a solution exists output m integers b_i ($0 \leq b_i \leq p-1$) which are the coefficients of $Q(x)$, separated by spaces. If there are multiple valid answers, output your favorite one.

Examples

standard input	standard output
2 1 1 2	1 7340031
4 2 1 0 1	1 0 7340032 0