Problem A. An Easy Geometry Problem

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

You are given an integer sequence $\{A_i\}$ of length n and a line $y = kx + b$ denoted by two integers k, b.

We say that an integer radius r of center i is **good** if and only if $i+r \leq n$, $i-r > 0$, and $A_{i+r} - A_{i-r} = kr+b$. We define the $\text{rad}(i)$ as the maximum integer r_0 so that for every $1 \leq r \leq r_0$, r is a good radius of center i.

You need to process queries of two types:

- $1\ l\ r\ v$: for every $l \leq i \leq r$, increase A_i by v;
- $2\ i$: calculate $\text{rad}(i)$.

Input

The first line of input contains four integers n, q, k, and b ($1 \leq n, q \leq 2 \times 10^5$, $|k|, |b| \leq 10^3$), denoting the length of the integer sequence, the number of queries, and the line.

The second line contains n integers $A_1, A_2, \ldots, A_n$ ($|A_i| \leq 10^3$), denoting the integer sequence.

The next q lines each contain a query, formatted as clarified in the statement. For each query of the first type, it is guaranteed that $1 \leq l \leq r \leq n$ and $|v| \leq 10^3$. For each query of the second type, it is guaranteed that $1 \leq i \leq n$.

Output

For every query of type 2, output a line denoting the answer.

Example

standard input	standard output
6 6 6 2 1 5 9 10 15 18 2 2 1 3 3 -3 2 2 1 3 4 3 2 3 2 4	1 0 2 0

Problem B. Counting Multisets

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	256 megabytes

For a multiset S consisting of non-negative integers, let $p(S)$ denote the number of distinct sequences obtained by permuting the elements of S.

For example, if $S = \{1, 1, 2\}$, then there are three distinct sequences: $\{1, 1, 2\}$, $\{1, 2, 1\}$, and $\{2, 1, 1\}$, so $p(S) = 3$.

For non-negative integers n, x, y, let $f(n, x, y)$ be the number of multisets S satisfying the following conditions: $|S| = n$, $\sum_{i \in S} i = x$, $\mathrm{OR}_{i \in S} i = y$, and $p(S)$ is odd.

Now, given non-negative integers $n, x, y_{\max}$, calculate $f(n, x, y)$ for any subset y of the binary representation of $y_{\max}$, modulo $10^9 + 7$.

Input

The first line contains a positive integer T ($1 \le T \le 100$), representing the number of test cases.

For the next T lines, each line contains three non-negative integers $n, x, y_{\max}$ ($1 \le n < 2^{30}$, $0 \le x < 2^{45}$, $0 \le y_{\max} < 2^{15}$), denoting each test case.

Let $\mathrm{pcnt}(x)$ represent the number of 1s in the binary representation of x. It is guaranteed that:

- the number of test cases with $\mathrm{pcnt}(y_{\max}) > 5$ does not exceed 30.
- the number of test cases with $\mathrm{pcnt}(y_{\max}) > 10$ does not exceed 4.

Output

For each test case, output one line with several integers. Specifically, for all subsets y of $y_{\max}$ in binary representation, output $f(n, x, y)$ in ascending order of y, separated by spaces.

Example

standard input	standard output
2 7 10 7 9 16 7	0 0 1 3 0 2 2 2 0 0 1 0 0 0 0 0

Problem C. Counting Strings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6.5 seconds
Memory limit:	256 megabytes

Given a string s of length n.

We say that a string t is **mentioned** in s if and only if there exist integers l, r, satisfying $1 \le l \le r \le n$, $\gcd(l, r) = 1$ and $s[l, r] = t$.

Here, $s[l, r]$ is defined as the string that is made by concatenating $s_l, s_{l+1}, \ldots, s_r$, and $\gcd(l, r)$ represents the greatest common divisor of l and r.

You should calculate the sum of the lengths of all distinct **non-empty** strings that are mentioned in s.

Input

The first line of the input contains the integer n ($1 \le n \le 100000$).

The second line contains the string s, consisting of only lowercase English letters.

Output

Print the sum of the lengths of all distinct **non-empty** strings that are mentioned on a single line.

Example

standard input	standard output
4 abca	14

Problem D. Bracket Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

A sequence is good when it can be represented by several () connected together.

Formally, a sequence s of length $2n$ is good when $s_1 = s_3 = \cdots = s_{2n-1} = ($ and $s_2 = s_4 = \cdots = s_{2n} =)$. In that case, we call n its **depth**.

Given a sequence s of length n which consists of (and). Let $f(l, r, k)$ be the number of good subsequences with depth k in the sequence t formed by $s_l, s_{l+1}, \ldots, s_r$.

You are given q queries, each query contains four numbers op, l, r, k.

- If $op = 1$, you need to calculate $f(l, r, k)$.
- If $op = 2$, you need to calculate $\sum_{l \le l' \le r' \le r} f(l', r', k)$.

Since the answer could be huge, you need to output the answer modulo 998244353.

Input

The first line contains two numbers n, q $(1 \le n \le 10^5, 1 \le q \le 10^6)$.

The second line contains the sequence s of length n.

The following q lines contain four numbers op, l, r, k $(op \in \{1, 2\}, 1 \le l \le r \le n, 1 \le k \le 20)$.

Output

Output q integers: the answer to each query, modulo 998244353.

Example

standard input	standard output
5 20 (()() 1 1 2 1 1 1 3 1 1 1 4 1 1 1 5 1 1 2 3 1 1 2 4 1 1 2 5 1 1 3 4 1 1 3 5 1 1 4 5 1 2 1 3 1 2 1 4 1 2 1 5 1 2 2 3 1 2 2 4 1 2 2 5 1 2 3 5 1 2 4 5 1 1 1 5 2 2 1 5 2	0 2 2 5 1 1 3 0 1 1 3 6 16 1 2 7 2 1 2 3

Problem E. Dominating Point

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You're given a complete directed graph G with n vertices. We call a vertex u **dominating** if for every $v \neq u$, there either exists an edge $(u \to v)$ or there exists a vertex w satisfying $(u \to w)$ and $(w \to v)$.

You now need to find 3 distinct dominating vertices of the given graph. If there are less than 3 dominating vertices, output `NOT FOUND`.

Input

The first line of input contains one integer n ($1 \leq n \leq 5000$).

The next n lines of input contain a binary string s_i each. There exists edge $(u \to v)$ if the v-th character of s_u is 1; otherwise, there is no such edge. It is guaranteed that exactly one of $s_{i,j} = 1$ and $s_{j,i} = 1$ holds for every $1 \leq i < j \leq n$ and $s_{i,i} = 0$ for every $1 \leq i \leq n$.

Output

The first line of output contains three integers a, b, c, denoting the answer you've found, or `NOT FOUND`, if there are not enough dominating vertices.

Examples

standard input	standard output
6 011010 000101 010111 100001 010100 100010	3 1 4
3 011 001 000	NOT FOUND
3 010 001 100	1 3 2

Problem F. An Easy Counting Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

You are given three integers k, p, x. Find the number of integer pairs (a, b) that satisfy the following conditions:

1. $0 \le b \le a < p^k$;
2. $\binom{a}{b} \equiv x \pmod{p}$.

Input

The only line of the input contains three integers: k ($1 \le k \le 2^{1000}$), p ($2 \le p \le 5000$) and x ($1 \le x < p$).

The integer k is given in its binary form, starting from the highest bit.

It is guaranteed that p is prime and that the first digit of k in the input is 1.

Output

Output a single integer – the answer modulo 998244353.

Examples

standard input	standard output
`1 7 5`	`2`
`1 43 17`	`17`
`1111111111 4999 1954`	`195378837`

Problem G. An Easy Math Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	128 megabytes

You are given an integer n. For each $p \le q$, where $pq \mid n$, denote $r = \frac{p}{q}$. Find the number of different values of r.

Input

The first line of the input contains one integer q ($1 \le q \le 2000$), denoting the number of test cases. The next q lines each contain one integer n ($1 \le n \le 10^{10}$).

Output

For each test case, print one line: an integer, the number of different values of r.

Example

standard input	standard output
10	1
1	2
2	2
3	3
4	2
5	5
6	2
7	4
8	3
9	5
10	

Problem H. Elimination Series Once More

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

2^n contestants take part in a programming contest. They are labeled from 1 to 2^n, and the contestant labeled i has a level of a_i. It is guaranteed that $a_1, a_2, \ldots, a_{2^n}$ forms a permutation of length 2^n (i.e., each a_i is an integer in the range $[1, 2^n]$, and $a_1, a_2, \ldots, a_{2^n}$ are pairwise distinct).

The contest takes the form of an elimination series. There are n rounds; each round will knock out exactly half of the contestants. In each round, let m be the current number of contestants, then the contestant with the smallest label (i.e., the label of each contestant remains unchanged throughout the contest) competes with the contestant with the second smallest label, the contestant with the third smallest label competes with the contestant with the fourth smallest label, and so on. In each competition, the contestant with a lower level is eliminated, while the contestant with a higher level will advance to the next round.

Little D is the contestant labeled x. He wants to use magic **before the contest** to win as many competitions as possible. Specifically, he can perform the following operation no more than k times: choose two contestants **other than himself** and swap their levels. Note that the operation can only be done **before the contest**.

For each $x = 1, 2, \ldots, 2^n$, print the maximum number of competitions he can win.

Input

The first line contains two integers n ($1 \le n \le 20$) and k ($0 \le k < 2^n$).

The second line contains 2^n integers $a_1, a_2, \ldots, a_{2^n}$. It is guaranteed that a is a permutation.

Output

For each $x = 1, 2, \ldots, 2^n$, print the maximum number of competitions Little D can win.

Example

standard input	standard output
`2 1` `1 2 3 4`	`0 1 1 2`

Problem I. Max GCD

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

You are given an array a of length n.

We define the value of an interval $[l, r]$ as

$$\max_{\substack{l \le i < j < k \le r \\ j-i \le k-j}} \gcd(a_i, a_j, a_k).$$

If $r - l \le 1$, the value is 0.

There will be q queries. In each query, you need to find the value of a particular interval.

Input

The first line contains two integers n and q ($3 \le n \le 1.5 \times 10^5$, $1 \le q \le 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$).

Each of the next q lines contains two integers l and r ($1 \le l \le r \le n$).

Output

For each query, output one line, an integer, the value of the interval.

Example

standard input	standard output
8 8 8 24 4 6 6 7 3 3 1 5 2 6 3 7 5 8 4 8 1 3 2 5 3 8	4 2 3 1 3 4 2 3

Problem J. Graph Changing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	32 megabytes

There are $10^9 + 1$ graphs with n vertices each. In the 0-th graph $G_0(V_0, E_0)$, there is an edge between vertices i and $i + 1$ for all $1 \le i < n$.

Denote the length of the shortest path between x and y in the graph i by $d_i(x, y)$. If x and y aren't reachable from each other, let $d_i(x, y)$ be -1. In the graph i ($i \ge 1$), there's an edge between x and y if and only if $d_{i-1}(x, y) \ge k$.

There are q queries. In each query you are given five numbers t, n, k, x, y. You need to output $d_t(x, y)$.

Input

The first line contains one number q ($1 \le q \le 10^6$).

In the following q lines, there are five numbers t, n, k, x, y ($0 \le t \le 10^9$, $2 \le n \le 10^9$, $1 \le k \le 10^9$, $1 \le x, y \le n$, $x \ne y$), denoting the query.

Output

For each query, output one line with one number $d_t(x, y)$.

Example

standard input	standard output
5	3
1 5 3 2 4	2
1 10 4 2 4	-1
2 10 5 2 4	1
1 3 2 1 3	-1
1 3 2 1 2	

Problem K. Penguins in Refrigerator

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

There are n penguins, where the width of the i-th penguin is w_i. They enter the refrigerator in an order specified by the permutation p_i: the p_i-th penguin enters the refrigerator at position i.

The refrigerator has a width of W, and its depth is considered infinite. After all these penguins have entered the refrigerator, they can swap positions if the sum of the widths of adjacent penguins does not exceed W. A penguin can swap positions multiple times.

Calculate how many different exit orders are possible from the refrigerator, as well as the lexicographically smallest order among all possible ways for the penguins to exit the refrigerator.

The refrigerator has only one door and operates as a "last in, first out" (LIFO) structure, which means the penguins will exit in the reverse order of their entry.

Input

The first line contains two positive integers, n and W ($1 \le n \le 10^6$, $1 \le W \le 10^9$), representing the number of penguins and the width of the refrigerator, respectively.

The second line contains a permutation of length n, denoted as p_i ($1 \le p_i \le n$), representing the order in which the penguins enter the refrigerator.

The third line contains n positive integers, w_i ($1 \le w_i \le W$), each of which represents the width of the penguin with the corresponding number i.

It is guaranteed that p_i is a permutation of $\{1, 2, \ldots, n\}$.

Output

For the first line, output an integer representing the ordinal number of the penguin coming out of the refrigerator, modulo $10^9 + 7$.

For the second line, output a permutation of length n, representing the lexicographically smallest order of penguins coming out of the refrigerator among all possible orders.

Examples

standard input	standard output
5 10 1 2 3 4 5 6 5 3 9 2	3 5 4 2 1 3
5 10 1 2 3 4 5 2 4 3 3 8	30 1 5 2 3 4

Problem L. Prism Palace

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Little D wants to build a prism palace using two parallel planes $A(z = 0)$ and $B(z = 10^9)$ that she has. To finish her build, she borrows a **convex** polygon P consisting of n points from Little N. She puts two identical copies of P on the two planes, one on each. The two polygons must be identical to P and must be able to coincide by translating the polygon on A along some vector $(d_x, d_y, 10^9)$ to the polygon on B without rotating.

Together, the two polygons form a prism-shaped palace between them. Let the projection area of the prism's sides perpendicularly onto plane A be $S_1, S_2, \cdots S_n$, and the probability of the multiset $\{S_i\}$ being **cool** be $f(r)$, assuming that (d_x, d_y) is chosen uniformly at random from the circle centered at $(0, 0)$ with a radius r, i.e., $d_x^2 + d_y^2 \le r^2$. Here, a multiset of real numbers S is **cool** if there exists an element x in S such that $\sum_{y \in S} y = 2x$. It can be proved that the limit $\lim_{r \to \infty} f(r)$ does exist, and you need to find this limit.

Input

The first line of input contains a single integer n $(3 \le n \le 2 \times 10^5)$, denoting the number of vertices of polygon P.

The next n lines each contain two integers (x_i, y_i) $(|x_i|, |y_i| \le 10^9)$, denoting the vertices of polygon P. The polygon P is formed by connecting (x_i, y_i) and $(x_{i \bmod n+1}, y_{i \bmod n+1})$ where $1 \le i \le n$. It is guaranteed that the polygon is convex, i.e., every inner angle of the polygon is smaller than π.

Output

The output contains only one real number, denoting your answer.

Let your output be u, and the answer be p. You'll get accepted if and only if $\frac{|u-p|}{\max(1,p)} \le 10^{-6}$.

Examples

standard input	standard output
3 0 0 1 0 0 1	1.000000000000000
4 0 0 0 1 1 1 1 0	0.000000000000000
4 0 0 0 3 1 2 1 1	0.500000000000000

Note

For the first test case, you can see that whatever the two polygons' positions are, the largest of the three projection areas is always the sum of the other two, so the answer is 100%, or 1.0.

Problem M. Random Variables

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

There are n random variables. Each of them is chosen uniformly at random from $[1, m] \cap \mathbb{Z}$.

Let occ_i, where $i \in [1, m] \cap \mathbb{Z}$, denote the number of times the value i occurs among the n variables. Let $M = \max\{occ_i | i \in [1, m] \cap \mathbb{Z}\}$.

For example, if $n = 5, m = 5$, the variables can be $\{4, 4, 3, 1, 5\}$ with probability $\frac{1}{3125}$. We have $occ_1 = 1$, $occ_2 = 0$, $occ_3 = 1$, $occ_4 = 2$, $occ_5 = 1$, $M = \max\{1, 0, 1, 2, 1\} = 2$.

Now you're given n and m; please work out the expected value of M. For convenience, let's denote the answer as $\mathrm{E}(M)$, then you only need to output $\mathrm{E}(M) \times m^n$ modulo p.

Input

The first line contains two positive integers T and p ($1 \le T \le 10^4$, $2 \le p \le 10^9 + 7$) — the number of test cases and the modulus.

The following T lines each contain two positive integers n and m ($1 \le n \le 1000$, $1 \le m \le 10^9$) — the number of random variables and the upper bound of each random variable.

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output

For each test case, print one line, a single integer — $\mathrm{E}(M) \times m^n$ modulo p.

Example

standard input	standard output
3 123456789 3 2 5 5 7 7	18 7145 2066323

Note

In the first test case, for results $\{1, 1, 2\}, \{1, 2, 1\}, \{1, 2, 2\}, \{2, 1, 1\}, \{2, 1, 2\}, \{2, 2, 1\}$, M equals 2; for results $\{1, 1, 1\}, \{2, 2, 2\}$, M equals 3. So, the total result is $1 \times 0 + 2 \times 6 + 3 \times 2 = 18$, which equals 18 modulo 123456789 .

Problem N. Python Program

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You need to calculate the output of a Python program.

To simplify the problem, the code satisfies the following constraints.

The program has exactly 5 lines. The first line of the code is

```
ans=0
```

The second and the third lines are of the form

```
for i in range(a,b,c):
    for j in range(d,e,f):
```

Here, i and j are loop variables. i and j can be any two different lowercase English letters. a, b, c, d, e, f can be integers without leading zeroes or loop variables that have been introduced before. That is, d, e and f may be i. The tokens c or f (or both) can be omitted. When omitted, they equal 1.

The fourth line of the code is of the form:

```
ans+=j
```

Here, j is the loop variable of the inner loop (note that it can be some letter other than `j`).

The fifth line of the code is

```
print(ans)
```

For any $c \neq 0$, the statement

```
for i in range(a,b,c):
```

in the Python language is interpreted as follows.

- If $c > 0$, i will take the values of $a, a+c, a+2c, \cdots$ until $a+kc$, where k is the maximum integer such that $a+kc < b$. If $a \geq b$, i won't take any value.
- If $c < 0$, i will take the values of $a, a+c, a+2c, \cdots$ until $a+kc$, where k is the maximum integer such that $a+kc > b$. If $a \leq b$, i won't take any value.

Input

5 lines, representing the code.

It is guaranteed that there are no extra spaces at the end of each line. However, there may be extra line feeds at the end of the input file. Please note that each tab is represented by 4 spaces in the input file.

It is guaranteed that $1 \leq a, b, |c|, d, e, |f| \leq 10^6$.

Output

One integer representing the output of the program.

Examples

standard input	standard output

<pre>ans=0
for a in range(1,3):
    for b in range(5,1,-2):
        ans+=b
print(ans)</pre>
	16

<pre>ans=0
for q in range(100,50,-1):
    for i in range(q,77,20):
        ans+=i
print(ans)</pre>
	2092

Note

For the first example, a will take the values of $1, 2$ in order, and for each value that a takes, b will take the values of $5, 3$, so the answer is $2 \times (5 + 3) = 16$.

You can test the programs with a Python interpreter.