Codeforces Round #729 (Div. 2)

A. Odd Set

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

You are given a multiset (i. e. a set that can contain multiple equal integers) containing $2n$ integers. Determine if you can split it into exactly n pairs (i. e. each element should be in exactly one pair) so that the sum of the two elements in each pair is **odd** (i. e. when divided by 2, the remainder is 1).

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \le t \le 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($1 \le n \le 100$).

The second line of each test case contains $2n$ integers $a_1, a_2, \dots, a_{2n}$ ($0 \le a_i \le 100$) — the numbers in the set.

Output

For each test case, print "Yes" if it can be split into exactly n pairs so that the sum of the two elements in each pair is **odd**, and "No" otherwise. You can print each letter in any case.

Example

input

```
5
2
2 3 4 5
3
2 3 4 5 5 5
1
2 4
1
2 3
4
1 5 3 2 6 7 3 4
```

output

```
Yes
No
No
Yes
No
```

Note

In the first test case, a possible way of splitting the set is $(2, 3)$, $(4, 5)$.

In the second, third and fifth test case, we can prove that there isn't any possible way.

In the fourth test case, a possible way of splitting the set is $(2, 3)$.

B. Plus and Multiply

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

There is an infinite set generated as follows:

- 1 is in this set.
- If x is in this set, $x \cdot a$ and $x + b$ both are in this set.

For example, when $a = 3$ and $b = 6$, the five smallest elements of the set are:

- 1,
- 3 (1 is in this set, so $1 \cdot a = 3$ is in this set),
- 7 (1 is in this set, so $1 + b = 7$ is in this set),
- 9 (3 is in this set, so $3 \cdot a = 9$ is in this set),
- 13 (7 is in this set, so $7 + b = 13$ is in this set).

Given positive integers a, b, n, determine if n is in this set.

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \le t \le 10^5$) — the number of test cases. The description of the test cases follows.

The only line describing each test case contains three integers n, a, b ($1 \le n, a, b \le 10^9$) separated by a single space.

Output

For each test case, print "Yes" if n is in this set, and "No" otherwise. You can print each letter in any case.

Example

input

```
5
24 3 5
10 3 6
2345 1 4
19260817 394 485
19260817 233 264
```

output

```
Yes
No
Yes
No
Yes
```

Note

In the first test case, 24 is generated as follows:

- 1 is in this set, so 3 and 6 are in this set;
- 3 is in this set, so 9 and 8 are in this set;
- 8 is in this set, so 24 and 13 are in this set.

Thus we can see 24 is in this set.

The five smallest elements of the set in the second test case is described in statements. We can see that 10 isn't among them.

C. Strange Function

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

Let $f(i)$ denote the minimum positive integer x such that x is **not** a divisor of i.

Compute $\sum_{i=1}^{n} f(i)$ modulo 10^9+7. In other words, compute $f(1)+f(2)+\dots+f(n)$ modulo 10^9+7.

Input

The first line contains a single integer t ($1 \le t \le 10^4$), the number of test cases. Then t cases follow.

The only line of each test case contains a single integer n ($1 \le n \le 10^{16}$).

Output

For each test case, output a single integer ans, where $ans = \sum_{i=1}^{n} f(i)$ modulo 10^9+7.

Example

input

```
6
1
2
3
4
10
10000000000000000
```

output

```
2
5
7
10
26
366580019
```

Note

In the fourth test case $n = 4$, so $ans = f(1)+f(2)+f(3)+f(4)$.

- 1 is a divisor of 1 but 2 isn't, so 2 is the minimum positive integer that isn't a divisor of 1. Thus, $f(1) = 2$.

- 1 and 2 are divisors of 2 but 3 isn't, so 3 is the minimum positive integer that isn't a divisor of 2. Thus, $f(2) = 3$.
- 1 is a divisor of 3 but 2 isn't, so 2 is the minimum positive integer that isn't a divisor of 3. Thus, $f(3) = 2$.
- 1 and 2 are divisors of 4 but 3 isn't, so 3 is the minimum positive integer that isn't a divisor of 4. Thus, $f(4) = 3$.

Therefore, $ans = f(1) + f(2) + f(3) + f(4) = 2 + 3 + 2 + 3 = 10$.

D. Priority Queue

time limit per test: 3 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

You are given a sequence A, where its elements are either in the form `+ x` or `-`, where x is an integer.

For such a sequence S where its elements are either in the form `+ x` or `-`, define $f(S)$ as follows:

- iterate through S's elements from the first one to the last one, and maintain a multiset T as you iterate through it.
- for each element, if it's in the form `+ x`, add x to T; otherwise, erase the smallest element from T (if T is empty, do nothing).
- after iterating through all S's elements, compute the sum of all elements in T. $f(S)$ is defined as the sum.

The sequence b is a subsequence of the sequence a if b can be derived from a by removing zero or more elements without changing the order of the remaining elements. For all A's subsequences B, compute the sum of $f(B)$, modulo $998\,244\,353$.

Input

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

Each of the next n lines begins with an operator `+` or `-`. If the operator is `+`, then it's followed by an integer x ($1 \le x < 998\,244\,353$). The i-th line of those n lines describes the i-th element in A.

Output

Print one integer, which is the answer to the problem, modulo $998\,244\,353$.

Examples

input

```
4
-
+ 1
+ 2
-
```

output

```
16
```

input

```
15
+ 2432543
-
+ 4567886
+ 65638788
-
+ 578943
-
-
+ 62356680
-
+ 711111
-
+ 998244352
-
-
```

output

```
750759115
```

Note

In the first example, the following are all possible pairs of B and $f(B)$:

- $B = \{\}$, $f(B) = 0$.
- $B = \{$`-`$\}$, $f(B) = 0$.
- $B = \{$`+ 1`, `-`$\}$, $f(B) = 0$.
- $B = \{$`-`, `+ 1`, `-`$\}$, $f(B) = 0$.
- $B = \{$`+ 2`, `-`$\}$, $f(B) = 0$.
- $B = \{$`-`, `+ 2`, `-`$\}$, $f(B) = 0$.
- $B = \{$`-`$\}$, $f(B) = 0$.
- $B = \{$`-`, `-`$\}$, $f(B) = 0$.
- $B = \{$`+ 1`, `+ 2`$\}$, $f(B) = 3$.
- $B = \{$`+ 1`, `+ 2`, `-`$\}$, $f(B) = 2$.

- $B=\{$`-`, `+ 1`, `+ 2`$\}$, $f(B)=3$.
- $B=\{$`-`, `+ 1`, `+ 2`, `-`$\}$, $f(B)=2$.
- $B=\{$`-`, `+ 1`$\}$, $f(B)=1$.
- $B=\{$`+ 1`$\}$, $f(B)=1$.
- $B=\{$`-`, `+ 2`$\}$, $f(B)=2$.
- $B=\{$`+ 2`$\}$, $f(B)=2$.

The sum of these values is 16.

E1. Abnormal Permutation Pairs (easy version)

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

This is the easy version of the problem. The only difference between the easy version and the hard version is the constraints on n. You can only make hacks if both versions are solved.

A permutation of $1, 2, \ldots, n$ is a sequence of n integers, where each integer from 1 to n appears exactly once. For example, $[2,3,1,4]$ is a permutation of $1,2,3,4$, but $[1,4,2,2]$ isn't because 2 appears twice in it.

Recall that the number of inversions in a permutation $a_1, a_2, \ldots, a_n$ is the number of pairs of indices (i, j) such that $i < j$ and $a_i > a_j$.

Let p and q be two permutations of $1, 2, \ldots, n$. Find the number of permutation pairs (p,q) that satisfy the following conditions:

- p is lexicographically smaller than q.
- the number of inversions in p is greater than the number of inversions in q.

Print the number of such pairs modulo mod. Note that mod may not be a prime.

Input

The only line contains two integers n and mod ($1 \le n \le 50$, $1 \le mod \le 10^9$).

Output

Print one integer, which is the answer modulo mod.

Example

input

```
4 403458273
```

output

```
17
```

Note

The following are all valid pairs (p,q) when $n=4$.

- $p=[1,3,4,2]$, $q=[2,1,3,4]$,
- $p=[1,4,2,3]$, $q=[2,1,3,4]$,
- $p=[1,4,3,2]$, $q=[2,1,3,4]$,
- $p=[1,4,3,2]$, $q=[2,1,4,3]$,
- $p=[1,4,3,2]$, $q=[2,3,1,4]$,
- $p=[1,4,3,2]$, $q=[3,1,2,4]$,
- $p=[2,3,4,1]$, $q=[3,1,2,4]$,
- $p=[2,4,1,3]$, $q=[3,1,2,4]$,
- $p=[2,4,3,1]$, $q=[3,1,2,4]$,
- $p=[2,4,3,1]$, $q=[3,1,4,2]$,
- $p=[2,4,3,1]$, $q=[3,2,1,4]$,
- $p=[2,4,3,1]$, $q=[4,1,2,3]$,
- $p=[3,2,4,1]$, $q=[4,1,2,3]$,
- $p=[3,4,1,2]$, $q=[4,1,2,3]$,
- $p=[3,4,2,1]$, $q=[4,1,2,3]$,
- $p=[3,4,2,1]$, $q=[4,1,3,2]$,
- $p=[3,4,2,1]$, $q=[4,2,1,3]$.

E2. Abnormal Permutation Pairs (hard version)

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

This is the hard version of the problem. The only difference between the easy version and the hard version is the constraints on n. You can only make hacks if both versions are solved.

A permutation of $1, 2, \ldots, n$ is a sequence of n integers, where each integer from 1 to n appears exactly once. For example, $[2, 3, 1, 4]$ is a permutation of $1, 2, 3, 4$, but $[1, 4, 2, 2]$ isn't because 2 appears twice in it.

Recall that the number of inversions in a permutation $a_1, a_2, \ldots, a_n$ is the number of pairs of indices (i, j) such that $i < j$ and $a_i > a_j$.

Let p and q be two permutations of $1, 2, \ldots, n$. Find the number of permutation pairs (p, q) that satisfy the following conditions:

- p is lexicographically smaller than q.
- the number of inversions in p is greater than the number of inversions in q.

Print the number of such pairs modulo mod. Note that mod may not be a prime.

Input

The only line contains two integers n and mod ($1 \le n \le 500$, $1 \le mod \le 10^9$).

Output

Print one integer, which is the answer modulo mod.

Example

input
4 403458273

output
17

Note

The following are all valid pairs (p, q) when $n = 4$.

- $p = [1, 3, 4, 2]$, $q = [2, 1, 3, 4]$,
- $p = [1, 4, 2, 3]$, $q = [2, 1, 3, 4]$,
- $p = [1, 4, 3, 2]$, $q = [2, 1, 3, 4]$,
- $p = [1, 4, 3, 2]$, $q = [2, 1, 4, 3]$,
- $p = [1, 4, 3, 2]$, $q = [2, 3, 1, 4]$,
- $p = [1, 4, 3, 2]$, $q = [3, 1, 2, 4]$,
- $p = [2, 3, 4, 1]$, $q = [3, 1, 2, 4]$,
- $p = [2, 4, 1, 3]$, $q = [3, 1, 2, 4]$,
- $p = [2, 4, 3, 1]$, $q = [3, 1, 2, 4]$,
- $p = [2, 4, 3, 1]$, $q = [3, 1, 4, 2]$,
- $p = [2, 4, 3, 1]$, $q = [3, 2, 1, 4]$,
- $p = [2, 4, 3, 1]$, $q = [4, 1, 2, 3]$,
- $p = [3, 2, 4, 1]$, $q = [4, 1, 2, 3]$,
- $p = [3, 4, 1, 2]$, $q = [4, 1, 2, 3]$,
- $p = [3, 4, 2, 1]$, $q = [4, 1, 2, 3]$,
- $p = [3, 4, 2, 1]$, $q = [4, 1, 3, 2]$,
- $p = [3, 4, 2, 1]$, $q = [4, 2, 1, 3]$.