A. We Got Everything Covered!

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

You are given two positive integers n and k.

Your task is to find a string s such that all possible strings of length n that can be formed using the first k lowercase English alphabets occur as a subsequence of s.

If there are multiple answers, print the one with the smallest length. If there are still multiple answers, you may print any of them.

Note: A string a is called a subsequence of another string b if a can be obtained by deleting some (possibly zero) characters from b without changing the order of the remaining characters.

Input

The first line of input contains a single integer t $(1 \leq t \leq 676)$ denoting the number of test cases.

Each test case consists of a single line of input containing two integers n $(1 \leq n \leq 26)$ and k $(1 \leq k \leq 26)$.

Output

For each test case, print a single line containing a single string s which satisfies the above property. If there are multiple answers, print the one with the smallest length. If there are still multiple answers, you may print any of them.

Standard Input	Standard Output
4 1 2 2 1 2 2 2 3	ab aa baab abcbac

Note

For the first test case, there are two strings of length 1 which can be formed using the first 2 lowercase English alphabets, and they are present in s as a subsequence as follows:

- `a` : `ab`
- `b` : `ab`

For the second test case, there is only one string of length 2 which can be formed using the first lowercase English alphabet, and it is present in s as a subsequence as follows:

- `aa` : `aa`

For the third test case, there are 4 strings of length 2 which can be formed using the first 2 lowercase English alphabets, and they are present in s as a subsequence as follows:

- `aa` : `baab`
- `ab` : `baab`
- `ba` : `baab`

- `bb` : `baab`

For the fourth test case, there are 9 strings of length 2 which can be formed using the first 3 lowercase English alphabets, and they are present in s as a subsequence as follows:

- `aa` : `abcbac`
- `ab` : `abcbac`
- `ac` : `abcbac`
- `ba` : `abcbac`
- `bb` : `abcbac`
- `bc` : `abcbac`
- `ca` : `abcbac`
- `cb` : `abcbac`
- `cc` : `abcbac`

,

B. A Balanced Problemset?

```
Input file:     standard input
Output file:    standard output
Time limit:     1.5 seconds
Memory limit:   256 megabytes
```

Jay managed to create a problem of difficulty x and decided to make it the second problem for Codeforces Round #921.

But Yash fears that this problem will make the contest highly unbalanced, and the coordinator will reject it. So, he decided to break it up into a problemset of n sub-problems such that the difficulties of all the sub-problems are a positive integer and their sum is equal to x.

The coordinator, Aleksey, defines the balance of a problemset as the GCD of the difficulties of all sub-problems in the problemset.

Find the maximum balance that Yash can achieve if he chooses the difficulties of the sub-problems optimally.

Input

The first line of input contains a single integer t $(1 \le t \le 10^3)$ denoting the number of test cases.

Each test case contains a single line of input containing two integers x $(1 \le x \le 10^8)$ and n $(1 \le n \le x)$.

Output

For each test case, print a single line containing a single integer denoting the maximum balance of the problemset Yash can achieve.

Standard Input	Standard Output
3 10 3 5 5 420 69	2 1 6

Note

For the first test case, one possible way is to break up the problem of difficulty 10 into a problemset having three problems of difficulties 4, 2 and 4 respectively, giving a balance equal to 2.

For the second test case, there is only one way to break up the problem of difficulty 5 into a problemset of 5 problems with each problem having a difficulty 1 giving a balance equal to 1.

,

C. Did We Get Everything Covered?

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

You are given two integers n and k along with a string s.

Your task is to check whether all possible strings of length n that can be formed using the first k lowercase English alphabets occur as a subsequence of s. If the answer is NO, you also need to print a string of length n that can be formed using the first k lowercase English alphabets which does not occur as a subsequence of s.

If there are multiple answers, you may print any of them.

Note: A string a is called a subsequence of another string b if a can be obtained by deleting some (possibly zero) characters from b without changing the order of the remaining characters.

Input

The first line of input contains a single integer $t\,(1 \le t \le 10^5)$, the number of test cases.

The first line of each test case contains 3 integers $n\,(1 \le n \le 26),\ k\,(1 \le k \le 26),\ m\,(1 \le m \le 1000)$, where n and k are the same as described in the input and m is the length of the string s.

The second line of each test case contains a single string s of length m, comprising only of the first k lowercase English alphabets.

It is guaranteed that the sum of m and the sum of n over all test cases does not exceed 10^6.

Output

For each test case, print YES if all possible strings of length n that can be formed using the first k lowercase English alphabets occur as a subsequence of s, else print NO.

If your answer is NO, print a string of length n that can be formed using the first k lowercase English alphabets which does not occur as a subsequence of s in the next line.

You may print each letter of YES or NO in any case (for example, YES, yES, YeS will all be recognized as a positive answer).

Standard Input	Standard Output
3 2 2 4 abba 2 2 3 abb 3 3 10 aabbccabab	YES NO aa NO ccc

Note

For the first test case, all possible strings (aa, ab, ba, bb) of length 2 that can be formed using the first 2 English alphabets occur as a subsequence of abba.

For the second test case, the string `aa` is not a subsequence of `abb`.

,

D. Good Trip

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

There are n children in a class, m pairs among them are friends. The i-th pair who are friends have a friendship value of f_i.

The teacher has to go for k excursions, and for each of the excursions she chooses a pair of children randomly, equiprobably and independently. If a pair of children who are friends is chosen, their friendship value increases by 1 for all subsequent excursions (the teacher can choose a pair of children more than once). The friendship value of a pair who are not friends is considered 0, and it does not change for subsequent excursions.

Find the expected value of the sum of friendship values of all k pairs chosen for the excursions (at the time of being chosen). It can be shown that this answer can always be expressed as a fraction $\frac{p}{q}$ where p and q are coprime integers. Calculate $p \cdot q^{-1} \bmod (10^9 + 7)$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 5 \cdot 10^4$). Description of the test cases follows.

The first line of each test case contains 3 integers n, m and k ($2 \le n \le 10^5$, $0 \le m \le \min\left(10^5, \frac{n(n-1)}{2}\right)$, $1 \le k \le 2 \cdot 10^5$) — the number of children, pairs of friends and excursions respectively.

The next m lines contain three integers each — a_i, b_i, f_i — the indices of the pair of children who are friends and their friendship value. ($a_i \neq b_i$, $1 \le a_i, b_i \le n$, $1 \le f_i \le 10^9$). **It is guaranteed that all pairs of friends are distinct**.

It is guaranteed that the sum of n and sum m over all test cases does not exceed 10^5 and the sum of k over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print one integer — the answer to the problem.

Standard Input	Standard Output
4 100 0 24 2 1 10 1 2 1 3 1 2 2 1 1 5 2 4 1 2 25 3 2 24	0 55 777777784 40000020

Note

For the first test case, there are no pairs of friends, so the friendship value of all pairs is 0 and stays 0 for subsequent rounds, hence the friendship value for all excursions is 0.

For the second test case, there is only one pair possible $(1, 2)$ and its friendship value is initially 1, so each turn they are picked and their friendship value increases by 1. Therefore, the total sum is $1 + 2 + 3 + \ldots + 10 = 55$.

For the third test case, the final answer is $\frac{7}{9} = 777\,777\,784 \bmod (10^9 + 7)$.

,

E. Space Harbour

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

There are n points numbered 1 to n on a straight line. Initially, there are m harbours. The i-th harbour is at point X_i and has a value V_i. **It is guaranteed that there are harbours at the points 1 and n.** There is exactly one ship on each of the n points. The cost of moving a ship from its current location to the next harbour is the product of the value of the nearest harbour to its left and the distance from the nearest harbour to its right. Specifically, if a ship is already at a harbour, the cost of moving it to the next harbour is 0.

Additionally, there are q queries, each of which is either of the following 2 types:

- $1\ x\ v$ — Add a harbour at point x with value v. It is guaranteed that before adding the harbour, there is no harbour at point x.
- $2\ l\ r$ — Print the sum of the cost of moving all ships at points from l to r to their next harbours. **Note that you just need to calculate the cost of moving the ships but not actually move them.**

Input

The first line contains three integers n, m, and q ($2 \le m \le n \le 3 \cdot 10^5$, $1 \le q \le 3 \cdot 10^5$) — the number of points, harbours, and queries, respectively.

The second line contains m distinct integers $X_1, X_2, \ldots, X_m (1 \le X_i \le n)$ — the position at which the i-th harbour is located.

The third line contains m integers $V_1, V_2, \ldots, V_m (1 \le V_i \le 10^7)$ — the value of the i-th harbour.

Each of the next q lines contains three integers. The first integer is t ($1 \le t \le 2$) — type of query. If $t = 1$, then the next two integers are x and v ($2 \le x \le n-1$, $1 \le v \le 10^7$) — first-type query. If $t = 2$, then the next two integers are l and r ($1 \le l \le r \le n$) — second-type query.

It is guaranteed that there is at least one second-type query.

Output

For every second-type query, print one integer in a new line — answer to this query.

Standard Input	Standard Output
8 3 4 1 3 8 3 24 10 2 2 5 1 5 15 2 5 5 2 7 8	171 0 15

Note

For the first type 2 query, the cost for ships at positions $2, 3, 4$ and 5 are $3 (3 \times 1)$, 0, $96 (24 \times 4)$ and $72 (24 \times 3)$ respectively.

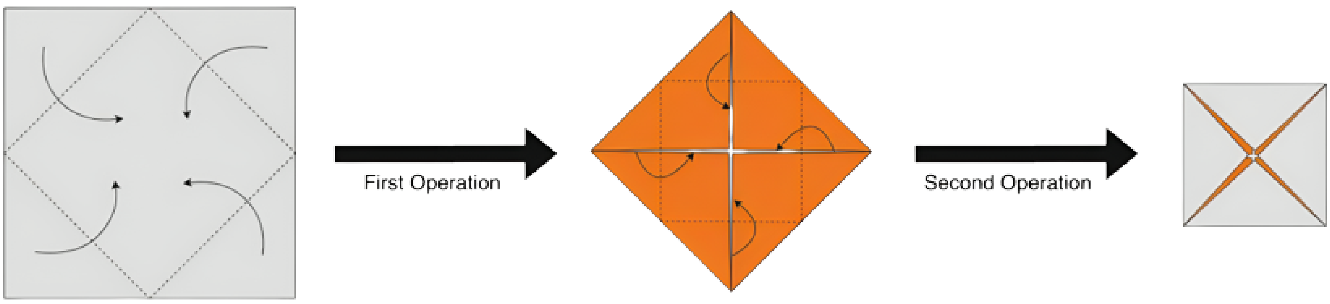
For the second type 2 query, since the ship at position 5 is already at a harbour, so the cost is 0.

For the third type 2 query, the cost for ships at position 7 and 8 are $15(15 \times 1)$ and 0 respectively.

F. Fractal Origami

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

You have a square piece of paper with a side length equal to 1 unit. In one operation, you fold each corner of the square to the center of the paper, thus forming another square with a side length equal to $\frac{1}{\sqrt{2}}$ units. By taking this square as a new square, you do the operation again and repeat this process a total of N times.



Performing operations for $N = 2$.

After performing the set of operations, you open the paper with the same side up you started with and see some crease lines on it. Every crease line is one of two types: a mountain or a valley. A mountain is when the paper folds outward, and a valley is when the paper folds inward.

You calculate the sum of the length of all mountain crease lines on the paper and call it M. Similarly, you calculate for valley crease lines and call it V. You want to find the value of $\frac{M}{V}$.

It can be proved that this value can be represented in the form of $A + B\sqrt{2}$, where A and B are rational numbers. Let this B be represented as an irreducible fraction $\frac{p}{q}$, your task is to print $p * inv(q)$ modulo $999\,999\,893$ **(note the unusual modulo)**, where $inv(q)$ is the modular inverse of q.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). Description of the test cases follows.

The only line of each test case contains an integer N ($1 \le N \le 10^9$), the number of operations you perform on the square paper.

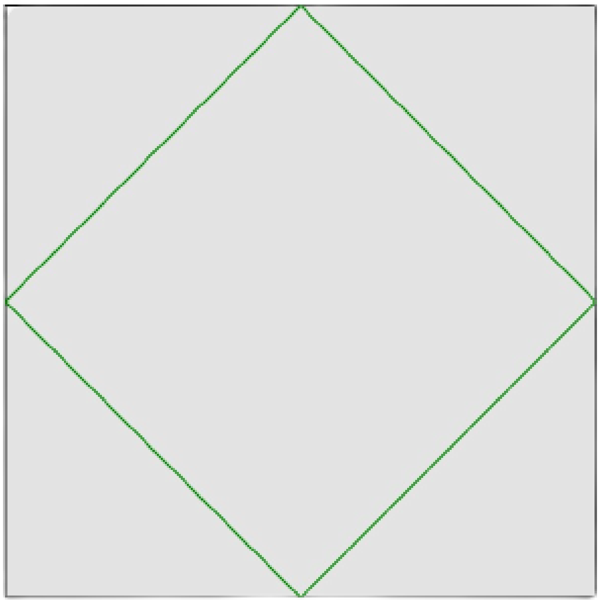
Output

For each test case, print on a new line the required answer.

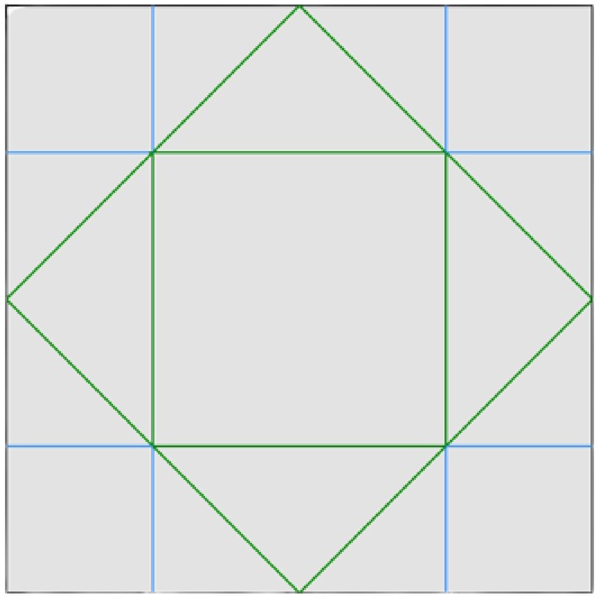
Standard Input	Standard Output
3 1 2 3	0 1 714285638

Note

The blue lines in the given figures represent mountain crease lines, and the green lines represent valley crease lines.



Crease lines after 1 operation ($\frac{M}{V} = 0$).



Crease lines after 2 operations ($\frac{M}{V} = \sqrt{2} - 1$).