A. 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`.

,

B. 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.

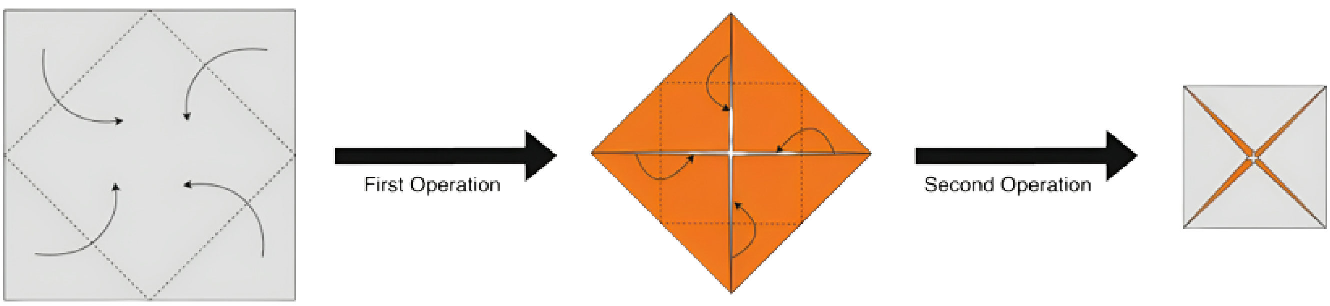
For the second type 2 query, since the ship at position 5 is already at a harbour, so the cost is 0.

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

C. 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.

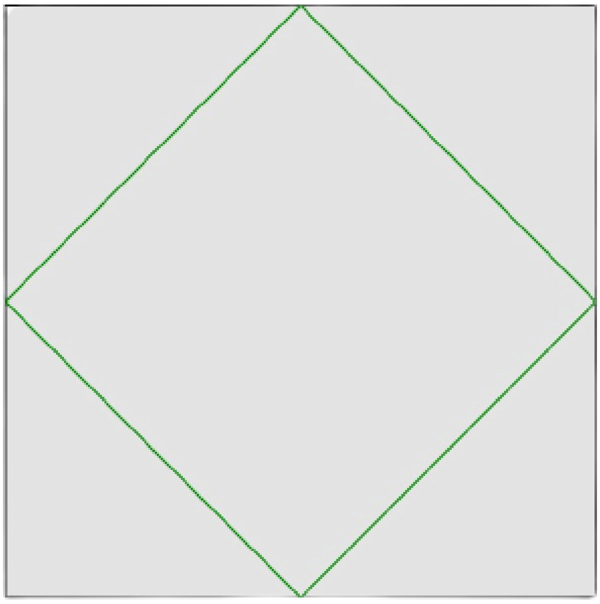
Output

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

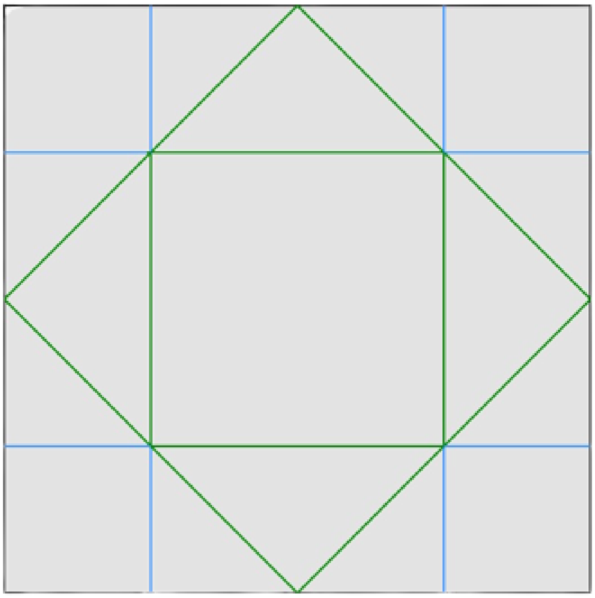
Standard Input	Standard Output
3 1 2 3	0 1 714285638

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 ($\frac{M}{V} = \sqrt{2} - 1$).

,

D. Balanced Subsequences

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

A sequence of brackets is called balanced if one can turn it into a valid math expression by adding characters `'+'` and `'1'`. For example, sequences `'(())()'`, `'()'`, and `'(()(()))'` are balanced, while `')('`, `'(()'`, and `'(()))('` are not.

A subsequence is a sequence that can be derived from the given sequence by deleting zero or more elements without changing the order of the remaining elements.

You are given three integers n, m and k. Find the number of sequences consisting of n `'('` and m `')'`, such that the longest balanced subsequence is of length $2 \cdot k$. Since the answer can be large calculate it modulo $1\,000\,000\,007$ ($10^9 + 7$).

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 3 \cdot 10^3$). Description of the test cases follows.

The first line of each test case contains three integers n, m and k ($1 \le n, m, k \le 2 \cdot 10^3$)

Output

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

Standard Input	Standard Output
3 2 2 2 3 2 3 3 2 1	2 0 4

Note

For the first test case `"()()"`, `"(())"` are the 2 sequences

For the second test case no sequence is possible.

For the third test case `")((()"`, `")(()("`, `")()(("`, `"()(("` are the 4 sequences.

,

E. Paper Cutting Again

```
Input file:     standard input
Output file:    standard output
Time limit:     3 seconds
Memory limit:   256 megabytes
```

There is a rectangular sheet of paper with initial height n and width m. Let the current height and width be h and w respectively. We introduce a xy-coordinate system so that the four corners of the sheet are $(0,0), (w,0), (0,h)$, and (w,h). The sheet can then be cut along the lines $x = 1, 2, \ldots, w-1$ and the lines $y = 1, 2, \ldots, h-1$. In each step, the paper is cut randomly along any one of these $h + w - 2$ lines. After each vertical and horizontal cut, the right and bottom piece of paper respectively are discarded.

Find the expected number of steps required to make the area of the sheet of paper strictly less than k. It can be shown that this answer can always be expressed as a fraction $\dfrac{p}{q}$ 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 ($1 \le t \le 57000$). Description of the test cases follows.

The first line of each test case contains 3 integers n, m, and k ($1 \le n, m \le 10^6$, $2 \le k \le 10^{12}$).

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

Output

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

Standard Input	Standard Output
4 2 4 10 2 4 8 2 4 2 2 4 6	0 1 833333342 250000003

Note

For the first test case, the area is already less than 10 so no cuts are required.

For the second test case, the area is exactly 8 so any one of the 4 possible cuts would make the area strictly less than 8.

For the third test case, the final answer is $\frac{17}{6} = 833\,333\,342 \bmod (10^9+7)$.

For the fourth test case, the final answer is $\frac{5}{4} = 250\,000\,003 \bmod (10^9+7)$.

,

F. Anti-Proxy Attendance

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

This is an interactive problem!

Mr. 1048576 is one of those faculty who hates wasting his time in taking class attendance. Instead of taking attendance the old-fashioned way, he decided to try out something new today.

There are n students in his class, having roll numbers 1 to n. He knows that **exactly 1 student is absent** today. In order to determine who is absent, he can ask some queries to the class. In each query, he can provide two integers l and r ($1 \le l \le r \le n$) and all students whose roll numbers are between l and r (inclusive) will raise their hands. He then counts them to determine if the roll number of the absent student lies between these values.

Things seemed fine until his teaching assistant noticed something — the students are dishonest! Some students whose roll numbers lie in the given range may not raise their hands, while some other students whose roll number does not lie in the given range may raise their hands. But the students don't want to raise much suspicion. So, only the following 4 cases are possible for a particular query (l, r) —

1. True Positive: $r - l + 1$ students are present and $r - l + 1$ students raised their hands.
2. True Negative: $r - l$ students are present and $r - l$ students raised their hands.
3. False Positive: $r - l$ students are present but $r - l + 1$ students raised their hands.
4. False Negative: $r - l + 1$ students are present but $r - l$ students raised their hands.

In the first two cases, the students are said to be answering honestly, while in the last two cases, the students are said to be answering dishonestly. The students can mutually decide upon their strategy, not known to Mr. 1048576. Also, the students do not want to raise any suspicion and at the same time, want to create a lot of confusion. So, their strategy always meets the following two conditions —

1. The students will never answer honestly 3 times in a row.
2. The students will never answer dishonestly 3 times in a row.

Mr. 1048576 is frustrated by this act of students. So, he is willing to mark at most 2 students as absent (though he knows that only one is). The attendance is said to be successful if the student who is actually absent is among those two. Also, due to limited class time, he can only ask up to $\lceil \log_{1.116} n \rceil - 1$ queries (weird numbers but okay). Help him complete a successful attendance.

Interaction

First read a line containing a single integer t ($1 \le t \le 2048$) denoting the number of independent test cases that you must solve.

For each test case, first read a line containing a single integer n ($3 \le n \le 10^5$). Then you may ask up to $\lceil \log_{1.116} n \rceil - 1$ queries.

To ask a query, print a single line in the format "`? l r`" (without quotes) $(1 \le l \le r \le n)$. Then read a single line containing a single integer x ($r - l \le x \le r - l + 1$) denoting the number of students who raised their hands corresponding to the query.

To mark a student as absent, print a single line in the format "`! a`" (without quotes) $(1 \le a \le n)$. Then read a single integer y ($y \in \{0, 1\}$). If the student with roll number a was absent, $y = 1$, else, $y = 0$. Note that this operation does not count as a query but you can do this operation at most 2 times.

To end a test case, print a single line in the format "`#`" (without quotes). Then you must continue solving the remaining test cases.

If you ask more queries than allowed or ask an invalid query, you will get the `Wrong answer` verdict.

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

After printing the answers, do not forget to output end of line and flush the output buffer. Otherwise, you will get the verdict `Idleness limit exceeded`. To flush the buffer, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- Read documentation for other languages.

Note that the grader for this problem is adaptive meaning that the answer may change depending on your queries but will always remain consistent with the constraints and the answer to the previous queries.

Input format for Hacks

The test cases for this problem use both non-adaptive and adaptive graders. You can use the non-adaptive grader for making hacks.

The first line of input contains a single integer t ($1 \le t \le 2048$).

The first line of each test case contains three integers g, n and x where $g = 1$ (to identify that this test case must use the non-adaptive grader), n ($3 \le n \le 10^5$) represents the number of students in the class and x ($1 \le x \le n$) represents the roll number of the student who is absent. You must ensure that the sum of n over all test cases does not exceed 10^5.

The second line of each test case contains a single string S ($1 \le |S| \le 120$, $S_i \in \{\mathtt{T}, \mathtt{F}\}$). This string represents the pattern of the truth sequence. If $S_{(i-1) \bmod |S| + 1} = \mathtt{T}$, the students will act honestly during the i-th query, otherwise they will act dishonestly. You must also ensure that there is no index i such that $S_{(i-1) \bmod |S|+1} = S_{i \bmod |S|+1} = S_{(i+1) \bmod |S|+1}$.

Standard Input	Standard Output
2	
5	? 1 4
3	? 3 5
2	? 2 2
1	? 1 3
2	? 3 3

0 1 0 2 0 1 6 6 2 2 0 1 1 0 0 0 1	? 3 3 ! 3 ? 2 4 ? 4 4 ! 2 # ? 1 6 ? 1 3 ? 4 6 ? 1 1 ? 3 3 ? 5 5 ! 3 ? 2 2 ? 4 4 ! 4 #

Note

For the first test case, the student with roll number 2 is absent and the truth sequence (see section for hacks) is `TFFTFTTF`. During execution of your solution, this test case will use a non-adaptive grader.

For the second test case, the student with roll number 4 is absent, and the truth sequence is `FFTFTTFT`. During the execution of your solution, in this test case your program will interact with an adaptive grader. So, the actual answer might be different depending on your queries but will always remain consistent with the responses to the previous queries.