Codeforces Round #716 (Div. 2)

A. Perfectly Imperfect Array

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

Given an array a of length n, tell us whether it has a non-empty subsequence such that the product of its elements is **not** a perfect square.

A sequence b is a subsequence of an array a if b can be obtained from a by deleting some (possibly zero) elements.

Input
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 length of the array a.

The second line of each test case contains n integers a_1, a_2, ..., a_n ($1 \le a_i \le 10^4$) — the elements of the array a.

Output
If there's a subsequence of a whose product isn't a perfect square, print "YES". Otherwise, print "NO".

Example

input
2 3 1 5 4 2 100 10000
output
YES NO

Note
In the first example, the product of the whole array (20) isn't a perfect square.

In the second example, all subsequences have a perfect square product.

B. AND 0, Sum Big

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Baby Badawy's first words were "AND 0 SUM BIG", so he decided to solve the following problem. Given two integers n and k, count the number of arrays of length n such that:

- all its elements are integers between 0 and $2^k - 1$ (inclusive);
- the bitwise AND of all its elements is 0;
- the sum of its elements is as large as possible.

Since the answer can be very large, print its remainder when divided by $10^9 + 7$.

Input
The first line contains an integer t ($1 \le t \le 10$) — the number of test cases you need to solve.

Each test case consists of a line containing two integers n and k ($1 \le n \le 10^5$, $1 \le k \le 20$).

Output
For each test case, print the number of arrays satisfying the conditions. Since the answer can be very large, print its remainder when divided by $10^9 + 7$.

Example

input
2 2 2 100000 20

output
4 226732710

Note
In the first example, the 4 arrays are:

- $[3, 0]$,
- $[0, 3]$,
- $[1, 2]$,
- $[2, 1]$.

C. Product 1 Modulo N

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

Now you get Baby Ehab's first words: "Given an integer n, find the longest subsequence of $[1, 2, \ldots, n-1]$ whose product is 1 modulo n." Please solve the problem.

A sequence b is a subsequence of an array a if b can be obtained from a by deleting some (possibly all) elements. The product of an empty subsequence is equal to 1.

Input
The only line contains the integer n ($2 \leq n \leq 10^5$).

Output
The first line should contain a single integer, the length of the longest subsequence.

The second line should contain the elements of the subsequence, **in increasing order**.

If there are multiple solutions, you can print any.

Examples

input
5
output
3 1 2 3

input
8
output
4 1 3 5 7

Note
In the first example, the product of the elements is 6 which is congruent to 1 modulo 5. The only longer subsequence is $[1, 2, 3, 4]$. Its product is 24 which is congruent to 4 modulo 5. Hence, the answer is $[1, 2, 3]$.

D. Cut and Stick

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

Baby Ehab has a piece of Cut and Stick with an array a of length n written on it. He plans to grab a pair of scissors and do the following to it:

- pick a range (l, r) and cut out every element a_l, a_{l+1}, ..., a_r in this range;
- stick some of the elements together in the same order they were in the array;
- end up with multiple pieces, where every piece contains some of the elements and every element belongs to some piece.

More formally, he partitions the sequence a_l, a_{l+1}, ..., a_r into subsequences. He thinks a partitioning is beautiful if for every piece (subsequence) it holds that, if it has length x, then no value occurs strictly more than $\lceil \frac{x}{2} \rceil$ times in it.

He didn't pick a range yet, so he's wondering: for q ranges (l, r), what is the minimum number of pieces he needs to partition the elements a_l, a_{l+1}, ..., a_r into so that the partitioning is beautiful.

A sequence b is a subsequence of an array a if b can be obtained from a by deleting some (possibly zero) elements. Note that it does **not** have to be contiguous.

Input
The first line contains two integers n and q ($1 \leq n, q \leq 3 \cdot 10^5$) — the length of the array a and the number of queries.

The second line contains n integers a_1, a_2, ..., a_n ($1 \leq a_i \leq n$) — the elements of the array a.

Each of the next q lines contains two integers l and r ($1 \leq l \leq r \leq n$) — the range of this query.

Output
For each query, print the minimum number of subsequences you need to partition this range into so that the partitioning is beautiful. We can prove such partitioning always exists.

Example

input
6 2 1 3 2 3 3 2 1 6 2 5
output
1 2

Note
In the first query, you can just put the whole array in one subsequence, since its length is 6, and no value occurs more than 3 times in it.

In the second query, the elements of the query range are $[3, 2, 3, 3]$. You can't put them all in one subsequence, since its length is 4, and 3 occurs more than 2 times. However, you can partition it into two subsequences: $[3]$ and $[2, 3, 3]$.

E. Baby Ehab's Hyper Apartment

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

This is an interactive problem.

Baby Ehab loves crawling around his apartment. It has n rooms numbered from 0 to $n-1$. For every pair of rooms, a and b, there's either a direct passage from room a to room b, or from room b to room a, but never both.

Baby Ehab wants to go play with Baby Badawy. He wants to know if he could get to him. However, he doesn't know anything about his apartment except the number of rooms. He can ask the baby sitter two types of questions:

- is the passage between room a and room b directed from a to b or the other way around?
- does room x have a passage towards any of the rooms s_1, s_2, ..., s_k?

He can ask at most $9n$ queries of the first type and at most $2n$ queries of the second type.

After asking some questions, he wants to know for every pair of rooms a and b whether there's a path from a to b or not. A path from a to b is a sequence of passages that starts from room a and ends at room b.

Input
The first line contains an integer t ($1 \leq t \leq 30$) — the number of test cases you need to solve.

Then each test case starts with an integer n ($4 \leq n \leq 100$) — the number of rooms.

The sum of n across the test cases doesn't exceed 500.

Output
To print the answer for a test case, print a line containing "3", followed by n lines, each containing a binary string of length n. The j-th character of the i-th string should be 1 if there's a path from room i to room j, and 0 if there isn't. The i-th character of the i-th string should be 1 for each valid i.

After printing the answer, we will respond with a single integer. If it's 1, you printed a correct answer and should keep solving the test cases (or exit if it is the last one). If it's -1, you printed a wrong answer and should terminate to get `Wrong answer` verdict. Otherwise, you can get an arbitrary verdict because your solution will continue to read from a closed stream.

Interaction
To ask a question of the first type, use the format:

- $1\ a\ b$ ($0 \leq a, b \leq n-1$, $a \neq b$).
- we will answer with 1 if the passage is from a to b, and 0 if it is from b to a.
- you can ask at most $9n$ questions of this type in each test case.

To ask a question of the second type, use the format:

- $2\ x\ k\ s_1\ s_2\ \ldots\ s_k$ ($0 \le x, s_i \le n-1$, $0 \le k < n$, $x \ne s_i$, elements of s are pairwise distinct).
- we will answer with 1 if there's a passage from x to any of the rooms in s, and 0 otherwise.
- you can ask at most $2n$ questions of this type in each test case.

If we answer with -1 instead of a valid answer, that means you exceeded the number of queries or made an invalid query. Exit immediately after receiving -1 and you will see `Wrong answer` verdict. Otherwise, you can get an arbitrary verdict because your solution will continue to read from a closed stream.

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

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

Hacks:

The first line should contain an integer t — the number of test cases.

The first line of each test case should contain an integer n ($4 \le n \le 100$) — the number of rooms.

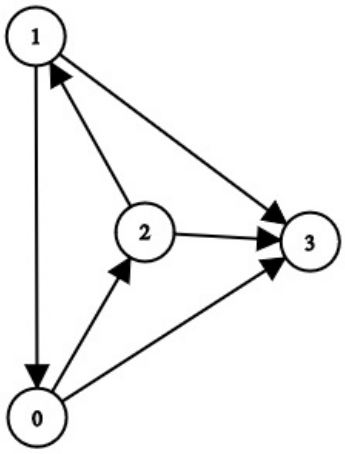
Each of the next n lines should contain a binary string of length n. The j-th character of the i-th string should be 1 if there's a passage from room i to room j, 0 otherwise. The i-th character of the i-th string should be 0.

Example

input
1 4 0 0 1 1 1
output
2 3 3 0 1 2 1 0 1 1 0 2 2 2 1 1 3 1111 1111 1111 0001

Note

In the given example:



The first query asks whether there's a passage from room 3 to any of the other rooms.

The second query asks about the direction of the passage between rooms 0 and 1.

After a couple other queries, we concluded that you can go from any room to any other room **except** if you start at room 3, and you can't get out of this room, so we printed the matrix:

1111
1111
1111
0001
The interactor answered with 1, telling us the answer is correct.
