

Problem A. PATH

File name: PATH.*
Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

Cho một cây bao gồm n đỉnh và $n - 1$ cạnh. Với mỗi cạnh theo thứ tự đầu vào, tìm số đường đi đơn phân biệt đi qua cạnh đó modulo 67.

Input

Dòng đầu tiên gồm một số nguyên dương n ($1 \leq n \leq 2 \times 10^5$) là số lượng đỉnh của cây. $n - 1$ dòng sau, dòng thứ i gồm hai số nguyên dương u_i, v_i là cạnh thứ i của cây.

Output

In ra $n - 1$ số, số thứ i là số lượng đường đi đơn phân biệt đi qua cạnh thứ i của cây theo thứ tự nhập vào modulo 67.

Scoring

- Subtask 1 (40%): $n \leq 200$.
- Subtask 2 (60%): Không có điều kiện gì thêm.

Example

Standard input	Standard output
3 1 2 2 3	2 2

Problem B. RECURIO

File name: RECURIO.*
Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

Cho hàm số $f(i)$ được định nghĩa bởi công thức sau:

$$\begin{cases} f(0) = 1 \\ f(n) = 1 \times f(n-1) + 2 \times f(n-2) + \dots + n \times f(0) \end{cases}$$

Tính giá trị của $f(n) \bmod 2718281828$ với t bộ dữ liệu đầu vào.

Input

Dòng đầu tiên gồm số nguyên dương t ($t \leq 10^5$) là số lượng câu hỏi.
 t dòng tiếp theo, mỗi dòng chứa một số tự nhiên n ($0 \leq n \leq 10^{18}$).

Output

In ra t dòng, mỗi dòng là giá trị $f(n) \bmod 2718281828$ tương ứng.

Scoring

- Subtask 1 (30%): $n \leq 2000, t = 1$.
- Subtask 2 (30%): $n \leq 2 \times 10^5$.
- Subtask 3 (40%): Không có điều kiện gì thêm.

Example

Standard input	Standard output
3	1
1	3
2	848560825
314159265353897932	

Problem C. ROTATEXT

File name: ROTATEXT.*
 Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 1024 megabytes

Cho hai xâu ký tự S và T cùng độ dài n , chỉ chứa c chữ cái đầu tiên trong bảng chữ cái tiếng Anh viết thường (từ ký tự thứ 1 đến ký tự thứ c). Một thao tác **shift** một ký tự là thay đổi ký tự đó thành ký tự kế tiếp trong bộ chữ cái c ký tự này.

Cụ thể, nếu ký tự là ký tự thứ i ($1 \leq i < c$), nó sẽ trở thành ký tự thứ $i + 1$. Nếu ký tự là ký tự thứ c , nó sẽ quay trở lại thành ký tự thứ 1 (ví dụ với $c = 3$: ' $a' \rightarrow 'b'$, ' $b' \rightarrow 'c'$, ' $c' \rightarrow 'a'$).

Có 3 loại truy vấn cần thực hiện:

- **1 l r x**: Thực hiện thao tác **shift** các ký tự từ vị trí l đến r của xâu S tổng cộng x lần.
- **2 l r x**: Thực hiện thao tác **shift** các ký tự từ vị trí l đến r của xâu T tổng cộng x lần.
- **3 l**: So sánh hai hậu tố $S[l \dots n]$ và $T[l \dots n]$. In ra $<$ nếu xâu S bé hơn, $>$ nếu xâu S lớn hơn, hoặc $=$ nếu hai xâu bằng nhau (theo thứ tự từ điển).

Input

Dòng đầu tiên gồm ba số nguyên dương n , c và q ($n, q \leq 2 \times 10^5$, $2 \leq c \leq 26$).

Dòng thứ hai chứa xâu S , dòng thứ ba chứa xâu T . Cả hai xâu đều có độ dài n và chỉ chứa c ký tự đầu tiên của bảng chữ cái tiếng Anh.

q dòng tiếp theo, mỗi dòng chứa một truy vấn theo định dạng đã mô tả. Trong các truy vấn loại 1 và 2, $1 \leq l \leq r \leq n$ và $1 \leq x < c$. Với truy vấn loại 3, $1 \leq l \leq n$.

Output

Với mỗi truy vấn loại 3, in ra kết quả tương ứng ($<$, $>$, hoặc $=$) trên một dòng.

Scoring

- Subtask 1 (30%): $n, q \leq 2000$.
- Subtask 2 (20%): Không có truy vấn loại 1 và loại 2.
- Subtask 3 (20%): $c = 3$.
- Subtask 4 (30%): Không có điều kiện gì thêm.

Example

Standard input	Standard output
3 4 3 abc abd 3 1 1 3 3 1 3 1	< =

Problem D. POWERGRID

File name: POWERGRID.*
Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

Bạn là chủ của tòa chung cư lớn nhất thế giới mang tên "Thành Phố Trên Mây". Tòa chung cư này là một hình chữ nhật gồm n tầng, mỗi tầng có m phòng được đánh số từ trái qua phải. Vào một ngày đẹp trời, hệ thống mạch điện của bạn bị lỗi một cách rất nghiêm trọng và bạn ngay lập tức gọi thợ điện.

Sau một hồi mày mò, bạn được biết rằng công tắc của một phòng điều khiển đèn của tất cả các phòng trên cùng tầng và cùng cột với phòng đó. Cụ thể hơn, khi bật công tắc của phòng (i, j) , tức phòng thứ j của tầng i , tất cả các phòng thuộc tầng i hoặc cột j đều bị ảnh hưởng (mỗi phòng bị ảnh hưởng đúng một lần). Khi một phòng bị ảnh hưởng, đèn của phòng đó đổi từ bật (1) thành tắt (0) hoặc tắt (0) thành bật (1). Tuy nhiên, các thợ điện cần thêm thời gian để chuẩn bị vật liệu để sửa chữa hệ thống điện của tòa chung cư này.

Tối hôm đó, cư dân trong tòa nhà bắt đầu than phiền vì đèn liên tục bật tắt bất thường, khiến họ không thể ngủ. Không còn cách nào khác, bạn quyết định tự mình can thiệp vào hệ thống bằng cách sử dụng những công tắc hiện có để tắt hết đèn để cư dân được ngủ.

Input

Dòng đầu tiên chứa số nguyên S ($1 \leq S \leq 5$) là số thứ tự subtask của test Dòng tiếp theo chứa số nguyên T ($1 \leq T \leq 10$) là số lượng trường hợp bạn cần xử lý. Mỗi trường hợp có định dạng như sau:

- Dòng đầu tiên chứa 2 số nguyên n, m ($n, m \leq 1000$)
- n dòng tiếp theo, mỗi dòng chứa m số, mỗi số là số 0 hoặc 1. Số thứ i trên dòng thứ j là trạng thái đèn của phòng (i, j) (0 là tắt, 1 là bật).

Dữ liệu đầu vào đảm bảo $\sum n \times m \leq 10^6$ qua các trường hợp.

Output

Với mỗi trường hợp:

- Nếu không có cách để tắt hết đèn, in ra NO
- Ngược lại, in ra $n + 1$ dòng. Dòng đầu in ra YES và n dòng tiếp theo, mỗi dòng chứa m số 0 hoặc 1 với số thứ i trên dòng thứ j là 1 nếu bạn muốn bật công tắc tại phòng (i, j) và 0 nếu ngược lại.

Scoring

- Subtask 1 (20%): $n \times m \leq 16$.
- Subtask 2 (20%): n, m đều là số chẵn, ma trận chỉ có đúng một số 1, $n, m \leq 100$.
- Subtask 3 (20%): n, m đều là số chẵn, $n, m \leq 100$.
- Subtask 4 (20%): n, m đều là số lẻ, $n, m \leq 100$.
- Subtask 5 (20%): Không có điều kiện gì thêm.

Example

Standard input	Standard output
1	YES
2	1 0
2 2	0 0
1 1	NO
1 0	
3 3	
1 0 1	
0 1 0	
1 0 1	

Note

Trong ví dụ đầu, việc bật công tắc của phòng $(1, 1)$ sẽ thay đổi trạng thái của các phòng $(1, 1), (1, 2), (2, 1)$ thành 0 và hoàn thành bài toán.

Trong ví dụ thứ hai, không tồn tại một cách để tắt hết đèn của các phòng.

Dưới đây là ví dụ minh họa cho một thao tác: với $n = m = 6$ và các đèn ban đầu là tắt, sau khi bật công tắc tại phòng $(2, 4)$, toàn bộ các phòng trên tầng 2 và cột 4 bị đảo trạng thái (ô màu vàng là bật, ô màu đen là tắt):

