

Problem A. Tổng - Tích

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

Xét một số nguyên dương x .

Gọi $f(x)$ là tích các chữ số của x và $g(x)$ là tổng các chữ số của x . Khi đó số x được gọi là đẹp nếu $f(x) \neq 0$ và $f(x)$ chia hết cho $g(x)$.

Cho một số nguyên dương n , hãy tìm một số x đẹp có n chữ số.

Input

Dòng đầu tiên chứa số nguyên dương T ($1 \leq T \leq 10^6$) — số lượng bộ test.

T dòng sau, mỗi dòng chứa một số nguyên dương n ($1 \leq n \leq 5 \times 10^6$) — số lượng chữ số của số cần tìm.

Gọi N là tổng n trong tất cả mọi test. Dữ liệu đảm bảo N không vượt quá 5×10^6 .

Output

Với mỗi test, in ra một số đẹp gồm n chữ số.

Scoring

Subtask 1 (20%): $n \leq 10$.

Subtask 2 (30%): $n = 2^k + 1$ với một số k nguyên không âm nào đó.

Subtask 3 (50%): Không có điều kiện gì thêm.

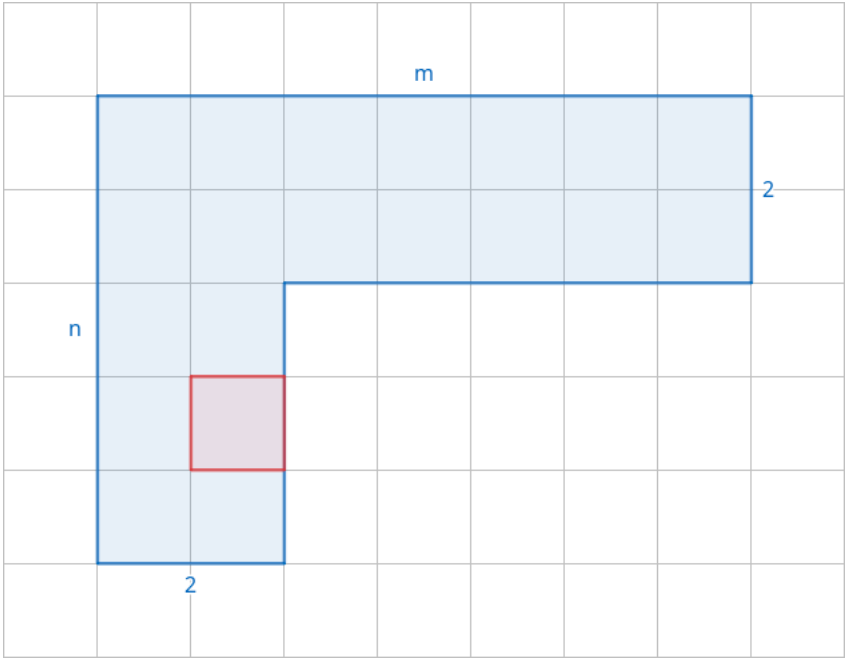
Example

standard input	standard output
6	9
1	63
2	167
3	6666
4	761
3	3789
4	

Problem B. Domino

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

Cho một hình chữ L với độ dài cạnh bên trái là n , độ dài cạnh trên là m , độ dày là 2. Các hàng được đánh số 1 đến n từ trên xuống dưới, các cột được đánh số 1 đến m từ trái sang phải. Có k ô vuông bị chặn có dạng (x, y) (Đảm bảo các ô nằm trong hình chữ L). Đây là ví dụ của hình chữ L có $n = 5, m = 7$, có ô $(4, 2)$ bị chặn:



Hãy đếm xem có bao nhiêu cách đặt các miếng domino 1×2 vào hình trên modulo 998244353.

Input

Dòng đầu tiên gồm 3 số n, m, k ($2 \leq n, m \leq 10^9, 0 \leq k \leq 10^4$).
 k dòng tiếp theo gồm 2 số (x, y) ($1 \leq x \leq n, 1 \leq y \leq m, x$ và y không cùng lớn hơn 2).

Output

Gồm một dòng là đáp án modulo 998244353.

Scoring

- Subtask 1(20%): $n, m \leq 10^6$
Subtask 2(40%): $k = 0$
Subtask 3(40%): Không có điều kiện thêm

Examples

standard input	standard output
2 2 0	7
3 2 1 1 1	10

Problem C. Sokoban

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

Bài toán này lấy ý tưởng từ một trò chơi nổi tiếng có tên là “Sokoban”.

Bob là người đã phá đảo màn chơi khó nhất của trò chơi Sokoban. Tuy nhiên, anh ấy đã tốn quá nhiều thời gian vào trò chơi này vì nó đòi hỏi tư duy sáng tạo và cả tính kiên nhẫn của người chơi nữa. Do đó, anh ta quyết định phát triển một mô hình có thể tự động giải một map Sokoban theo yêu cầu của người dùng. Vì đây là một bài toán cực khó, nên Bob muốn bắt đầu từ một mô hình cơ bản trước: giải trò chơi “One-box Sokoban”. Luật chơi của trò chơi này như sau:

Cho một lưới ô vuông có kích thước $n \times m$ gồm n hàng, m cột. Các hàng được đánh số từ 1 đến n từ trên xuống dưới, các cột được đánh số từ 1 đến m từ trái sang phải, ô nằm giao giữa hàng i và cột j được gọi là ô (i, j) . Một số ô trong lưới là ô bị chặn, không thể di chuyển vào được.

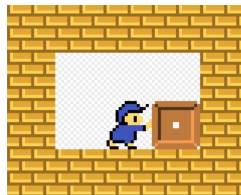
Có một con Sokoban đứng ở ô (x_S, y_S) và một cái thùng nằm ở ô (x_B, y_B) . Nhiệm vụ của bạn là điều khiển con Sokoban sao cho nó đưa được thùng đến ô (x_G, y_G) ($1 \leq x_S, x_B, x_G \leq n$, $1 \leq y_S, y_B, y_G \leq m$).

Đảm bảo 3 ô (x_S, y_S) , (x_B, y_B) và (x_G, y_G) đều không bị chặn và đôi một phân biệt. Ở mỗi bước, bạn có thể cho Sokoban di chuyển lên trên, xuống dưới, sang trái hoặc sang phải một ô, miễn sao Sokoban luôn nằm ở trong bảng và không đi vào ô bị chặn. Trên đường đi, nếu gặp thùng, Sokoban sẽ “đẩy” thùng dịch chuyển theo hướng đi của mình nếu thùng không đi ra khỏi bảng cũng như không đi vào ô bị chặn. Để hiểu rõ hơn, hãy xem xét ví dụ dưới đây:

Với $n = 4$, $m = 5$, các ô ở biên của bảng bị chặn, các ô còn lại thì không, Sokoban đứng ở ô $(3, 2)$, thùng nằm ở ô $(3, 3)$, ta có trạng thái giống như ở hình dưới đây:



Ở trạng thái trên, ta có thể điều khiển Sokoban “đẩy” thùng sang bên phải. Khi đó, Sokoban sẽ đứng ở ô $(3, 3)$, thùng nằm ở ô $(3, 4)$:



Ở trạng thái này, Sokoban không thể đẩy thùng sang phải tiếp được nữa vì ô $(3, 5)$ là ô bị chặn.

Vì Bob muốn kiểm thử mô hình của mình, nên anh ấy đã nhờ bạn giúp. Yêu cầu: cho một map One-box Sokoban, hãy xác định liệu bạn có thể hoàn thành nhiệm vụ (đưa được thùng đến ô (x_G, y_G)) hay không.

Lưu ý: trò chơi “One-box Sokoban” chỉ có một cái thùng duy nhất, khác với trò Sokoban thông thường.

Input

Mỗi test gồm có nhiều testcase. Dòng đầu tiên là số nguyên dương T — số lượng testcase ($1 \leq T \leq 10$).

T nhóm dòng tiếp theo, mỗi nhóm dòng biểu diễn một testcase và có dạng như sau:

- Dòng đầu tiên gồm hai số nguyên dương n, m — kích thước của lưới ô vuông ($2 \leq n, m \leq 10^5$, $n \times m \leq 10^5$).
- n dòng tiếp theo, mỗi dòng gồm m kí tự. Kí tự thứ j ở dòng thứ i là $a_{i,j}$ biểu diễn trạng thái bắt đầu của ô (i, j) . $a_{i,j}$ chỉ có thể là một trong 5 kí tự '.', '#', 'S', 'B', 'G'. Trong đó:
 - Nếu $a_{i,j}$ là '#' thì ô (i, j) là ô bị chặn.
 - Nếu $a_{i,j}$ là 'S' thì ô (i, j) là ô bắt đầu của Sokoban, hay $(i, j) = (x_S, y_S)$.
 - Nếu $a_{i,j}$ là 'B' thì ô (i, j) là ô bắt đầu của thùng, hay $(i, j) = (x_B, y_B)$.
 - Nếu $a_{i,j}$ là 'G' thì ô (i, j) là ô mà bạn cần điều khiển Sokoban đẩy thùng đến, hay $(i, j) = (x_G, y_G)$.
 - Nếu $a_{i,j}$ là '.' thì ô (i, j) không có gì cả.

Dữ liệu đảm bảo có đúng 1 kí tự 'S', 1 kí tự 'B' và 1 kí tự 'G' trong $n \times m$ kí tự này.

Output

Gồm T dòng, dòng thứ i là câu trả lời của testcase thứ i : in ra "YES" nếu bạn có thể hoàn thành nhiệm vụ, ngược lại in ra "NO" (không có dấu ngoặc kép, không phân biệt chữ thường, chữ hoa).

Example

standard input	standard output
2 9 7##S .##G...### ##..#.. ##.##.. ##B##.. #..##.. #..#### #..#... 9 7##S ###G...### ##..#.. ##.##.. ##B##.. #..##.. #..#### #..#...	YES NO

Note

Giải thích test ví dụ:

- Ở testcase thứ nhất, một cách để điều khiển Sokoban hoàn thành nhiệm vụ được biểu diễn tại tệp 'sokoban.gif' trên hệ thống thi.

- Ở testcase thứ hai, mọi thứ đều giống testcase thứ nhất, ngoại trừ việc ô $(2, 1)$ bị chặn. Điều này khiến cho Sokoban không thể hoàn thành nhiệm vụ được.

Lưu ý: Có 40% số test tương ứng với 40% số điểm có $n \times m \leq 1000$.

Problem D. Mã linh tinh

Input file: **standard input**
Output file: **standard output**
Time limit: 7 seconds
Memory limit: 256 megabytes

Cho một mảng N số nguyên dương A và số nguyên C . Xử lý Q truy vấn thuộc một trong ba dạng sau:

- 1 1 r v : Thực hiện phép gán $A_i := A_i * v$ với mọi $l \leq i \leq r$.
- 2 1 r v : Thực hiện phép gán $A_i := A_i + v$ với mọi $l \leq i \leq r$.
- 3 1 r : Định nghĩa $f(x, y) = \sum_{i=x}^y A_i$. Với mỗi $0 \leq K \leq 4$, tính tổng $(f(x, y) + C)^K$ với mọi cặp (l, r) thỏa mãn $l \leq x \leq y \leq r$ theo modulo 1 000 000 007. Nói cách khác, với mỗi K thỏa mãn $0 \leq K \leq 4$, in ra:

$$\left(\sum_{x=l}^r \sum_{y=x}^r \left(C + \sum_{i=x}^y A_i \right)^K \right) \mod 1\,000\,000\,007$$

Chú ý: Thí sinh không cần trả lời đúng với mọi K để được điểm. Xem phần **Scoring** để biết thêm chi tiết.

Input

Dòng đầu tiên là ba số N, Q, C ($1 \leq N, Q \leq 100\,000, 0 \leq C < 1\,000\,000\,007$).

Dòng thứ hai là N số nguyên A_i ($1 \leq A_i < 1\,000\,000\,007$).

Sau dòng thứ hai là Q dòng, dòng thứ i biểu diễn cho truy vấn thứ i và thuộc một trong ba dạng:

- 1 1 r v ($1 \leq l \leq r \leq n, 1 \leq v < 1\,000\,000\,007$): Truy vấn dạng 1
- 2 1 r v ($1 \leq l \leq r \leq n, 1 \leq v < 1\,000\,000\,007$): Truy vấn dạng 2
- 3 1 r ($1 \leq l \leq r \leq n$): Truy vấn dạng 3

Output

Với mỗi truy vấn dạng 3, in ra một dòng gồm 5 số nguyên thuộc $[0, 1\,000\,000\,007)$, là kết quả của các phép tính.

Scoring

Tên subtask	Tổng điểm	Giới hạn N, Q	Truy vấn dạng 2
Subtask 1	8%	$N, Q \leq 500$	Không
Subtask 2	8%	$N, Q \leq 500$	Có
Subtask 3	21%	$N, Q \leq 50\,000$	Không
Subtask 4	21%	$N, Q \leq 50\,000$	Có
Subtask 5	21%	$N, Q \leq 100\,000$	Không
Subtask 6	21%	$N, Q \leq 100\,000$	Có

Chú ý: Thí sinh không cần giải đúng với mọi K để được điểm dương. Chẳng hạn, để được tính điểm cho $K = 3$, thí sinh chỉ cần giải đúng với $K = 3$ trong mọi truy vấn 3. Cụ thể, tỉ lệ điểm với giới hạn N, Q và giá trị K trả lời đúng được cố định như sau:

	$N, Q \leq 500$	$N, Q \leq 50\,000$	$N, Q \leq 100\,000$
$k = 0$	4%	4%	4%
$k = 1$	4%	6%	6%
$k = 2$	4%	12%	12%
$k = 3$	2%	8%	8%
$k = 4$	2%	12%	12%

Ví dụ, nếu thí sinh trả lời đúng với $N, Q \leq 50\,000$ và $k \in \{0, 3\}$, sẽ được $4\% + 2\% + 4\% + 8\%$ số điểm của bài.

Với mỗi dòng trong đầu ra, thí sinh vẫn cần phải in ra đúng 5 số thuộc khoảng $[0, 1\,000\,000\,007)$ để được điểm.

Examples

standard input	standard output
3 3 4 4 3 3 1 1 3 4 2 1 3 4 3 3 3	1 20 400 8000 160000
4 4 4 2 7 6 6 3 3 3 1 1 1 5 1 1 4 1 3 2 4	1 10 100 1000 10000 6 87 1395 24507 463539
4 4 10 1 2 9 6 3 3 4 1 1 3 6 2 2 4 1 3 2 3	3 60 1242 26580 586482 3 166 10838 761344 55145522
4 4 3 4 8 6 7 3 3 4 3 2 3 1 2 4 2 3 2 4	3 35 437 5825 82097 3 37 491 6973 104723 6 156 4702 160452 5995894
10 10 10 6 12 10 1 5 13 15 2 1 1 2 7 9 8 1 2 10 15 1 2 8 1 1 7 9 9 3 5 9 1 2 6 12 2 9 10 3 3 3 5 2 5 6 8 3 1 3	15 46905 208478625 779393492 303109245 6 8880 17640600 3367720 431372103 6 14118 44289708 877035991 608288387