

## شوشو شکمو

• محدودیت زمانی: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

شوشو به شکمو بودن معروف است. ما هم تصمیم گرفته‌ایم تا با او یک بازی کنیم. تعدادی جعبه شکلات در یک ردیف داریم و هر بار به شوشو یک دستور جدید می‌دهیم.

۱. این جعبه را بنداز دور.

۲. یک جعبه با  $x$  تعداد شکلات بعد این جعبه قرار بده.

۳. برو سراغ جعبه بعدی.

۴. برو سراغ جعبه قبلی.

شوشو از اولین جعبه شروع می‌کند. او باید هر جعبه‌ای که در مسیر دیده است را به خاطر داشته باشد. در نهایت یک دنباله از جعبه‌ها را دارد و باید برای هر عضو دنباله باید ببیند که اولین جعبه‌ای که شکلات بیشتری را دارد و در دنباله جلوتر از این جعبه است چیست و به آن جعبه دو امتیاز مثبت می‌دهد و اولین جعبه‌ای که شکلات کمتری دارد و در دنباله عقب‌تر از این جعبه است چیست و به آن جعبه یک امتیاز مثبت می‌دهد. در نهایت جعبه‌ای که در دنباله بیشترین امتیاز را دارد به شوشو تقدیم می‌شود (اگر امتیاز دو جعبه برابر بود جعبه‌ای که شکلات بیشتری دارد مد نظر است). از آن جایی که شوشو شکمو و فقط به فکر شکم خودش است، قبل از شمردن تعداد شکلات‌ها آن‌ها را می‌خورد و شما باید بگویید که آن جعبه چه تعداد شکلات دارد.

دقت داشته باشید که هر جعبه‌ای که در شروع داده می‌شود یکتاست و یکتایی جعبه‌ها با تعداد شکلات‌هایشان تعیین نمی‌شود. یک جعبه ممکن است چندین بار در مسیر دیده شود، در این صورت ذات جعبه‌های دیده‌شده یک چیز است و جعبه‌ای که چندین بار دیده شده به عنوان جعبه‌های متمایز نباید در نظر گرفته شود.

## ورودی

در خط اول ورودی‌های  $n$  و  $m$  را با فاصله خواهید داشت که  $n$  تعداد جعبه‌های اولیه است و  $m$  تعداد دستورها.

در  $n$  خط اول تعداد شکلات‌های هر جعبه داده می‌شود.

در  $m$  خط دوم در هر دستور در یک خط به شما داده می‌شود که می‌تواند فقط شامل یک عدد یا دو عدد با فاصله باشد. اگر شامل یک عدد بود، تنها نوع دستور ۱ یا ۳ یا ۴ را مشخص کرده است. اگر شامل دو عدد بود، دستور شماره ۲ است و عدد دوم برابر با  $x$  می‌باشد.

توجه داشته باشید که در صورت دور ریختن جعبه، شوشو به طور خودکار سراغ جعبه بعدی می‌رود و آن جعبه را دیده است؛ اگر جعبه بعدی وجود نداشته و جعبه قبلی وجود داشت شوشو سراغ جعبه قبلی می‌رود و آن جعبه را دیده است. همچنین در صورت اضافه کردن جعبه جدید، آن جعبه را ندیده است و جای شوشو تغییر نکرده است. اگر امکان رفتن به جعبه بعدی یا قبلی وجود نداشته (به علت وجود نداشتن جعبه) شوشو باید سر جای خودش بماند.

تضمین می‌شود که هیچ حالتی وجود ندارد که در حین اعمال تغییرات و حرکت بین جعبه‌ها هیچ جعبه‌ای وجود نداشته باشد و حتما حداقل یک جعبه وجود دارد.

$$1 \leq n, m \leq 2 * 10^5$$

## خروجی

باید تعداد شکلات‌های جعبه نهایی را در خروجی بدهید.

## مثال

### ورودی نمونه ۱

5  
6  
9  
1  
3  
3  
3

## خروجی نمونه ۱

6

جعبه‌های ما به ترتیب زیر هستند:

5, 6, 9, 1

از جعبه ۵ شروع کرده و هر بار یکی جلو می‌رویم و کل جعبه‌ها را می‌بینیم. اولین عنصر کوچکتر از ۵ و سمت چپش وجود ندارد و اولین عنصر بزرگتر سمت راستش جعبه با عدد ۶ است. اولین کوچکتر ۶ جعبه با عدد ۵ است و اولین بزرگتر جعبه با عدد ۹ است. اولین کوچکتر ۹ جعبه با عدد ۶ است و اولین بزرگتر وجود ندارد. برای ۱ هم هیچ کدام وجود ندارند. در نهایت امتیاز جعبه با ۶ شکلات برابر با ۳ است. امتیاز جعبه با ۵ شکلات عدد ۱ است. امتیاز جعبه با ۹ شکلات عدد ۲ است. جعبه با ۶ شکلات بیشترین امتیاز را دارد.

## ورودی نمونه ۲

4 6  
3  
1  
5  
1  
3  
3  
3  
4  
4

1

خروجی نمونه ۲

5

ورودی نمونه ۳

5 9

7

7

0

9

0

2 3

2 3

3

1

3

3

4

2 0

1

خروجی نمونه ۳

7

در این جا چیزی که باید دقت شود این است که بالاترین امتیاز معادل با ۶ است. امتیازهای جعبه‌ها به ترتیب امتیازهای زیر است:

0, 0, 1, 6, 1, 6, 0

دو جعبه‌ای که امتیاز ۶ به آن‌ها نسبت داده شده در آرایه اصلی در واقع یک جعبه هستند که در دو زمان متفاوت دیده شده‌اند.