QUEUE

Bài 1:

Do những ngày hè quá nóng bức và nhàm chán nên Tide đã nghĩ ra một trò chơi khá thú vị với queue. Ban đầu trong queue có 5 số 1, 2, 3, 4, 5 với mỗi lượt chơi Tide sẽ xóa phần tử ở đầu queue và cho 2 phần tử đó xuống cuối của queue và cứ tiếp tục cho đến khi Tide cảm thấy mệt và không chơi được nữa.

Ví dụ tại lượt chơi thứ nhất trạng thái của queue là 1, 2, 3 ,4 ,5

Tại lượt chơi thứ 2 trạng thái của queue là 2, 3, 4, 5, 1, 1

Các bạn hãy giúp Tide xác định xem số đầu tiên của queue tại lượt chơi thứ N nhé.

Input

Gồm duy nhất số N (N<=10^9)

Output

Kết quả tìm được.

Example

**Test 1:  
Input:**

1  
**Output:**

1

**Test 2:  
Input:**6  
**Output:**1

Bài 2:

Ngân hàng BIG-Bank mở một chi nhánh ở Bucharest và được trang bị một máy tính hiện đại với các công nghệ mới nhập, C2#,VC3+ ... chỉ chuối mỗi cái là không ai biết lập trình.

Họ cần một phần mềm mô tả hoạt động của ngân hàng như sau: mỗi khách hàng có một mã số là số nguyên K, và khi đến ngân hàng giao dịch, họ sẽ nhận được 1 số P là thứ tự ưu tiên của họ. Các thao tác chính như sau

0 Kết thúc phục vụ

1 K P Thêm khách hàng K vào hàng đợi với độ ưu tiên P

2 Phục vụ người có độ ưu tiên cao nhất và xóa khỏi danh sách hàng đợi

3 Phục vụ người có độ ưu tiên thấp nhất và xóa khỏi danh sách hàng đợi.

Tất nhiên là họ cần bạn giúp rồi.

### Input

Mỗi dòng của input là 1 yêu cầu, và chỉ yêu cầu cuối cùng mới có giá trị là 0. Giả thiết là khi có yêu cầu 1 thì không có khách hàng nào khác có độ ưu tiên là P.

K<=10^6, và P<= 10^7.Một khách hàng có thể yêu cầu phục vụ nhiều lần và với các độ ưu tiên khác nhau.

### Output

Với mỗi yêu cầu 2 hoặc 3, in ra trên 1 dòng mã số của khách hàng được phục vụ tương ứng. Nếu có yêu cầu mà hàng đợi rỗng, in ra số 0.

### Sample

Input :

2

1 20 14

1 30 3

2

1 10 99

3

2

2

0

Ouput:

0

20

30

10

0

STACK

Cho một biểu thức hậu tố với số hạng là các số nguyên dương và ba toán tử **+, -, \***. Hãy tính giá trị của biểu thức hậu tố.

Ví dụ: biểu thức hậu tố: **2 3 4 + \* 5 – 2 2 \* +** có giá trị là 13.

**Dữ liệu nhập:**

– Gồm một dòng thể hiện biểu thức hậu tố, mỗi số hạng là một số nguyên dương trong phạm vi từ 1 đến 100. Giữa hai số hạng, hoặc giữa hai toán tử, hoặc giữa số hạng và toán tử, cách nhau một khoảng trắng. Chiều dài biểu thức không quá 100 ký tự. Dữ liệu đề bài cho đảm bảo biểu thức hậu tố là hợp lệ. Trong quá trình tính toán đảm bảo trị tuyệt đối các giá trị trung gian không vượt quá 109.

**Dữ liệu xuất:**

– Là giá trị của biểu thức hậu tố.

**Lưu ý:** dùng hàm gets() hay getline() để đọc chuỗi.

**Ví dụ**

* input:  2 3 4 + \* 5 - 2 2 \* +
* output:  13

HASHTABLE

WORDFIND - Trò chơi tìm chữ

Dữ liệu vào: standard input

Dữ liệu ra: standard output

Giới hạn thời gian: 4.0 giây

Giới hạn bộ nhớ: 128 megabyte

Sau kì thi VOI căng thẳng, Misaki quyết định thư giãn bằng cách chơi trò chơi tìm chữ. Cách chơi như sau: Cho bảng chữ cái kích thước NxM, cần tìm trên bảng Q dãy các chữ cái được tạo thành bằng 1 trong các cách nối như sau:  
  1. Nối các chữ cái nằm trên cùng 1 dòng.

  2. Nối các chữ cái nằm trên cùng 1 cột.

  3. Nối các chữ cái nằm trên đường chéo từ phía trên bên trái xuống phía dưới bên phải.

  4. Nối các chữ cái nằm trên đường chéo từ phía trên bên phải xuống phía dưới bên trái.

Ban đầu, các chữ cái trên bảng rất dễ tìm. Nhưng sau đó độ khó của game tăng lên, kích thước của bảng, số lượng chữ lớn hơn ban đầu rất nhiều, Misaki không thể tìm hết các chữ. Bạn hãy viết chương trình giúp Misaki tìm các chữ chưa tìm được nhé!

**Yêu cầu:**Cho bảng chữ cái NxM và Q chữ cần tìm, bạn hãy tìm vị trí chữ cái đầu tiên của dãy có cách nối tạo thành chữ đó.

**Dữ liệu vào:**

- Dòng đầu chứa 3 số nguyên N,M,Q (1 ≤ N,M ≤ 200 | 1 ≤ Q ≤ 1000).

  - N dòng tiếp theo, mỗi dòng chứa M chữ cái, chữ cái aijlà chữ cái ở hàng i và cột j của bảng.

  - Dòng thứ N+2 là dòng trống.

  - Q dòng tiếp theo chứa một chữ là chữ cần tìm.

**Dữ liệu ra:**

  Gồm Q dòng tương ứng với Q chữ cần tìm, mỗi dòng chứa 3 số nguyên u,v,d lần lượt là số dòng, số cột và cách nối tương ứng với cách đánh số ở đề bài. Nếu có nhiều vị trí thỏa yêu cầu, in ra vị trí có số dòng nhỏ nhất. Nếu có nhiều vị trí trên dòng thỏa yêu cầu, in ra vị trí có số cột nhỏ nhất. Nếu vẫn còn nhiều cách nối ở 1 ô, in ra cách nối có số thứ tự nhỏ nhất. Nếu không tìm được vị trí thỏa yêu cầu thì in ra -1.

**Lưu ý:**Các chữ cái sau khi nối sẽ được đọc từ trái qua phải (với cách nối 1,3,4) hoặc từ trên xuống dưới (với cách nối 2).

**Ví dụ**

* **input**

8 8 3  
nawdawdh  
dtasoyjl  
hauzjhvn  
nwqcktyv  
twdaodur  
tixdfdjm  
ytopjvef  
azxgbmhr  
  
ntucoder  
dhnt  
abcd

**output**

1 1 3  
2 1 2  
-1