**LKNIGHT – MÃ ĐI TUẦN**

Không còn việc gì làm, Lê Minh quay lại với bài toán chưa giải được của mình, bài toán mã đi tuần trên bàn cờ N\*N.

Mã đi tuần (hay còn gọi là hành trình của quân mã), là bài toán về việc di chuyển một quân mã trên bàn cờ N\*N. Quân mã được đặt ở trên một bàn cờ trống và phải di chuyển theo nguyên tắc của cờ vua để đi qua mỗi ô trên bàn cờ đúng một lần. Trong bài toán này, ta tạm bỏ qua việc một ô chỉ được đi qua đúng một lần.

Để thuận tiện, chúng ta sẽ đánh số các nước đi mà quân mã có thể thực hiện bằng các số nguyên từ 1 đến 8 như sau (M thể hiện vị trí ban đầu của quân mã):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 8 |  | 1 |  |
| 7 |  |  |  | 2 |
|  |  | M |  |  |
| 6 |  |  |  | 3 |
|  | 5 |  | 4 |  |

Bất chợt, Lê Minh nhận ra rằng mình đã quên không ghi lại bất kỳ thông tin nào về lời giải mình đang xây dựng dở vì ham chơi. Tuy nhiên, là một người có trí nhớ phi thường (có thể nhớ được tất cả các đề bài, lời giải cũng như các test khó của tất cả các bài mình đã làm), Lê Minh đã thuộc chính xác tất cả các nước đi của quân mã mà mình đã thực hiện. Khi đi theo các nước đi này, quân mã sẽ không đi ra ngoài bàn cờ. Tuy nhiên, Lê Minh không nhớ vị trí xuất phát của quân mã (dĩ nhiên, nhớ một dãy số dài luôn đơn giản hơn nhớ 1 2 con số vô nghĩa).

Nhiệm vụ của bạn là giúp Lê Minh đếm xem sau các nước đi mà cậu đã thực hiện, quân mã có thể ở bao nhiêu vị trí khác nhau, nếu quân mã có thể đi vào một vị trí nhiều lần.

**Dữ liệu vào:** Gồm 2 dòng:

* Dòng thứ nhất ghi 2 số nguyên N và K (8≤N≤1000, 1≤K≤1000), lần lượt là kích thước bàn cờ và số nước đi mà Lê Minh đã thực hiện.
* Dòng thứ hai ghi K chữ số, mỗi chữ số trong khoảng từ 1 đến 8, thể hiện các bước di chuyển mà Lê Minh đã thực hiện (không có dấu cách ở giữa các chữ số)

**Dữ liệu ra:** Gồm 1 số nguyên duy nhất là kết quả của bài toán

|  |  |
| --- | --- |
| **LKNIGHT.INP** | **LKNIGHT.OUT** |
| 8 2  11 | 24 |

Đánh số các hàng từ 1 đến N từ trên xuống, đánh số các cột từ 1 đến N từ trái sang phải.

Các vị trí của quân mã sau 2 nước đi có thể là:

(1,3), (1,4), (1,5), (1,6), (1,7), (1,8), (2,3), (2,4), (2,5), (2,6), (2,7), (2,8), (3,3), (3,4), (3,5), (3,6), (3,7), (3,8), (4,3), (4,4), (4,5), (4,6), (4,7), (4,8).

Chú ý:

Trong 60% test, N ≤ 300 và K ≤ 600.

Công ty đường sắt Lê Minh quyết định theo kịp thời đại và bước đầu làm quen với hệ thống mạng liên thành phố. Bởi vì đầu máy xe lửa hoạt động kém hiệu quả nên chỉ có một tuyến đường thẳng kết nối các thành phố. Việc thiếu một hệ thống máy tính dành cho công việc đặt chỗ còn gây thêm trở ngại khác.

Để đơn giản, ta có thể xem như xe lửa chạy ngang qua c thành phố được đánh số từ 1 đến c ( thành phố bắt đầu được đánh số 1 và thành phố cuối cùng được đánh số c). Có tất cả s chỗ ngồi trên xe lửa. Việc chuyên chở số hành khách nhiều hơn số chỗ ngồi trên xe lửa là không được phép.

Hệ thống máy tính liên tiếp nhận được một số yêu cầu đặt chỗ và cần được hoàn thành. Một yêu cầu được chấp nhận khi trên đoạn đường được yêu cầu đặt chỗ có đủ số ghế trống dành cho số hành khách đặt chỗ. Ngược lại, yêu cầu bị hủy bỏ. Không chấp nhận một phần của yêu cầu, tức là không chấp nhận việc hành khách chỉ đi trên một phần đoạn đường đã đặt chỗ hoặc số hành khách lên xe ít hơn so với số lượng đã đăng ký đặt chỗ. Sau khi chấp nhận một yêu cầu nào đó thì phải cập nhật lại số lượng ghế đã có hành khách ngồi. Các yêu cầu được gửi đến theo thứ tự thời gian.

**Yêu cầu:** Viết chương trình tính toán xem các yêu cầu được chấp nhận hay bị huỷ bỏ.

### Dữ liệu vào:

Trên dòng đầu tiên ghi 3 số nguyên c,s,r lần lượt cho biết số lượng thành phố mà đường sắt có đi ngang qua, tổng số ghế ngồi trên xe lửa và số lượng yêu cầu đặt chỗ   
(1<=*n*<=100 000, 1<=*s*<=100 000, 1<=*m*<=100 000).

Trên m dòng tiếp theo mô tả các yêu cầu đặt chỗ liên tiếp theo thứ tự thời gian.

Trên dòng thứ I +1 mô tả yêu cầu thứ i.

Một mô tả bao gồm 3 số nguyên u, v và c cho biết chỉ số của thành phố khởi hành, chỉ số thành phố kết thúc và số hành khách sẽ lên xe

### Dữ liệu ra: Gồm m dòng

### Dòng thứ i ghi đúng một kí tự cho biết yêu cầu thứ i có được chấp nhận hay không:

* T - Nếu yêu cầu thứ i được chấp nhận.
* N – Nếu ngược lại.

### Ví dụ:

|  |  |
| --- | --- |
| **KOL.INP** | **KOL.OUT** |
| 4 6 4  1 4 2  1 3 2  2 4 3  1 2 3 | T  T  N  N |