

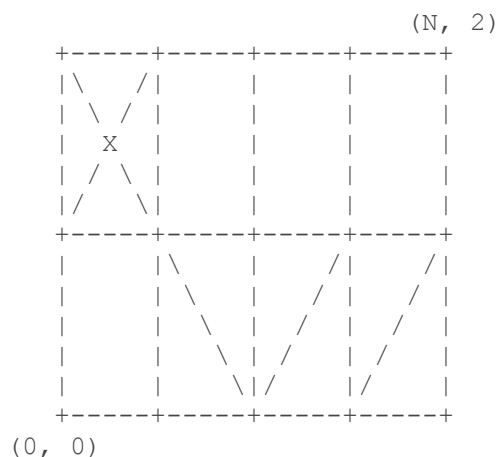
## City Inspection

Chef Rams nổi tiếng với những công thức nấu ăn tuyệt vời và cách sử dụng ngôn ngữ đầy màu sắc của anh ta, và anh ta muốn mở một nhà hàng mới trong thành phố ManiLand.

Thành phố của ManiLand có mô hình cấu trúc mạng lưới thông thường. Thành phố có ba **con đường chính** trải dài từ đông sang tây và  **$N+1$  đường nhỏ** từ nam xuống bắc. Con đường chính đầu tiên trải dài từ tọa độ  $(0, 0)$  tới tọa độ  $(N, 0)$ , đường chính thứ hai trải dài từ  $(0, 1)$  tới  $(N, 1)$ , và đường chính thứ ba trải dài từ  $(0, 2)$  tới  $(N, 2)$ . Đường nhỏ thứ  $i$  ( $0 \leq i \leq N$ ) trải dài từ  $(i, 0)$  tới  $(i, 2)$ . Thêm nữa, nó còn có những **đường trung chuyển** bắt chéo từ một điểm  $(x, y)$  nào đó tới điểm  $(x+1, y+1)$  hoặc chéo từ một điểm  $(x, y)$  nào đó tới điểm  $(x+1, y-1)$  ( $x$  và  $y$  là các số nguyên). Tất cả các con đường đều có hai chiều.

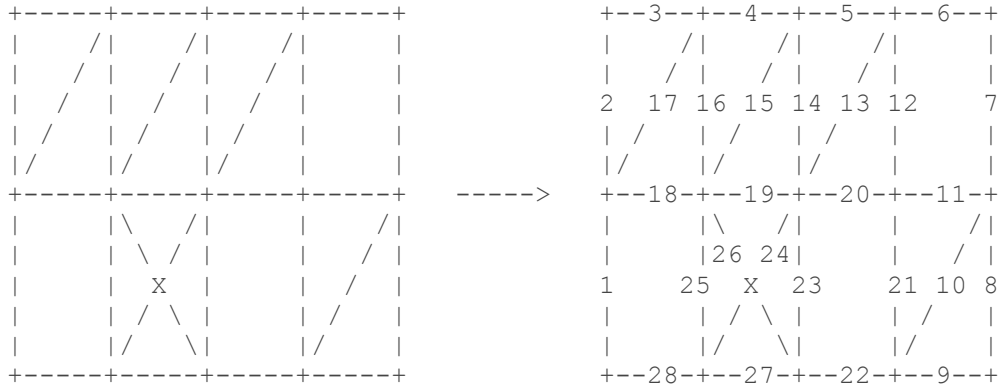
Chúng ta định nghĩa một block là một ô hình vuông giới hạn bởi các giao lộ  $(x, y)$ ,  $(x, y+1)$ ,  $(x+1, y+1)$ ,  $(x+1, y)$  với  $x$  và  $y$  là các số nguyên. ManiLand có chiều dài là  $N$  block và rộng là  $2$  block.

Hình dưới đây cho thấy một sơ đồ điền hình của ManiLand (lưu ý rằng các đường trung chuyển trên cùng một block cũng giao nhau).



Trạm cảnh sát nằm ở giao lộ  $(0, 0)$ . Hiện nay, cảnh sát hằng ngày thường đi tuần tra ở các con đường và các đoạn đường mà họ đi qua. Tuy nhiên, họ phát hiện ra rằng bất cứ khi nào họ đi tuần tra, họ luôn đi qua một con đường (hay một đoạn đường) nhiều hơn một lần.

Một sơ đồ được gọi là "**inspection-friendly**" nếu một người có thể từ trụ sở cảnh sát đi qua các đoạn đường đúng một lần (không tính các giao lộ), và quay lại trụ sở. Sơ đồ ở trên thì không "inspection-friendly", nhưng những cái dưới đây thì có:



(Lưu ý rằng việc mỗi con đường được đi qua đúng một lần trong khi những giao lộ được đi qua hai lần vẫn là chấp nhận được.)

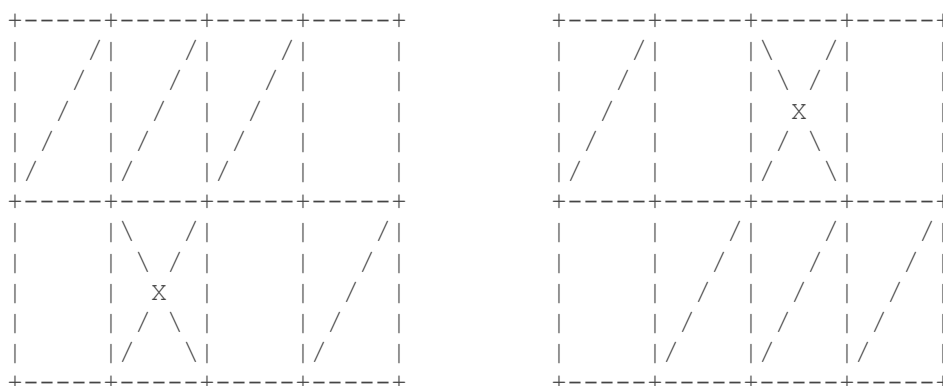
Hiện giờ, chính phủ muốn làm cho ManiLand trở nên “inspection-friendly” vì cảnh sát không thích phải đi qua một con đường hai lần và đôi khi end bỏ qua vài đường. Chi tiết là chính phủ dự định sẽ phá hủy một số đường trung chuyển và xây những cái mới. Tuy nhiên, những đường chính và đường nhỏ phải sẽ được giữ nguyên.

Chi phí của việc xây dựng và phá bỏ một đường trung chuyển bắt chéo từ  $(x, y)$  tới  $(x+1, y+1)$  lần lượt là  $H_b$  và  $H_d$  và chi phí để xây dựng và phá bỏ một đường trung chuyển bắt chéo từ  $(x, y)$  tới  $(x+1, y-1)$  lần lượt là  $L_b$  và  $L_d$ . Chính phủ muốn biết **tổng chi phí nhỏ nhất** cần thiết để làm cho ManiLand trở nên “inspection-friendly”, và **số lượng phương án** để đạt được tổng chi phí là nhỏ nhất. Những con đường trung chuyển phải tuân theo luật *ManiLand Publick Works and Highways*:

- Tất cả những con đường trung chuyển phải bắt chéo từ  $(x, y)$  tới  $(x+1, y+1)$  hoặc từ  $(x, y)$  tới  $(x+1, y-1)$ ;
- Chúng không thể xây dựng ngoài mạng lưới; và
- Không có hai đường trung chuyển có thể kết nối cùng một cặp giao lộ.

Lưu ý rằng thứ tự các con đường được xây/phá bỏ là không quan trọng; hai phương án được coi là khác nhau chỉ khi thành quả sơ đồ ManiLand cuối cùng là khác nhau.

Trong hình đầu tiên ở trên, giả định  $H_b = 20$ ,  $H_d = 15$ ,  $L_b = 4$ ,  $L_d = 31$ , thì chi phí nhỏ nhất là **106** và có **2** phương án để làm được như vậy, được biểu diễn trong hình dưới đây:



Chính phủ muốn thuê Chef Rams để tính toán 2 giá trị này cho họ (họ không muốn thuê những lập trình viên thực sự vì tính tình họ rất hay bất thường). Họ cũng lấy đây làm điều kiện cần thiết cho việc cung cấp giấy phép cho đề đạt về nhà hàng của Chef Rams. Chef Rams rất muốn mở nhà hàng nhưng không biết chút gì về lập trình, cầu cứu bạn giúp đỡ. Hãy giúp Chef Rams!

Hiện giờ, ManiLand rất dài (có nghĩa là  $N$  rất lớn), nhưng may thay, sơ đồ hiện tại có thể đoán được. Toàn bộ sơ đồ chiều dài  $N$  chứa một đoạn chiều dài  $K$  (tất là có  $2K$  blocks) lặp lại  $N/K$  lần. (Lưu ý rằng bạn không cần phải duy trì cấu trúc này khi đang cố làm cho thành phố trở nên “inspection-friendly”).

#### Dữ liệu vào:

- Dòng đầu tiên của dữ liệu vào chứa một số nguyên  $T$  là số lượng bộ dữ liệu. Mô tả của  $T$  bộ dữ liệu như sau:
- Dòng đầu của mỗi bộ dữ liệu gồm sáu số nguyên  $H_b, H_d, L_b, L_d, N$  và  $K$  là chi phí để xây/phá như miêu tả ở trên, và độ dài của sơ đồ và độ dài của cấu trúc được lặp lại. Dòng thứ hai và thứ ba cùng chứa một chuỗi độ dài  $K$  gồm các ký tự ‘/’, ‘\’, ‘X’ và ‘.’, mỗi cái đại diện cho một block. ‘/’ và ‘\’ đại diện cho một block với một đường trung chuyển. ‘X’ đại diện cho block với hai đường trung chuyển, và ‘.’ đại diện cho một block không có đường trung chuyển nào. Hãy nhìn vào dữ liệu vào mẫu để có thêm thông tin.

#### Dữ liệu ra:

- Với mỗi bộ dữ liệu, xuất ra một dòng duy nhất chứa hai số nguyên là chi phí nhỏ nhất và số lượng phương án để đạt được chi phí này. Vì số lượng phương án để đạt được tổng chi phí này có thể rất lớn, chỉ xuất ra **số dư của đáp án  $10^9+7$** .
- Nếu không thể làm nó trở thành “inspection-friendly”, xuất ra “a kitchen nightmare” (không dấu ngoặc kép).

#### Ràng buộc:

- $1 \leq T \leq 10^5$
- $1 \leq K \leq 10^5$
- $1 \leq N \leq 10^{15}$
- Tổng các các giá trị  $K$  trong một tập tin dữ liệu tối đa là  $10^5$
- $N$  chia hết cho  $K$
- $1 \leq H_b, H_d, L_b, L_d \leq 1000$

#### Ví dụ:

##### Input:

```
3
20 15 4 31 4 4
X...
.\//
1 3 4 10 2 1
X
/
11 11 11 11 1 1
```

**Output:**

106 2

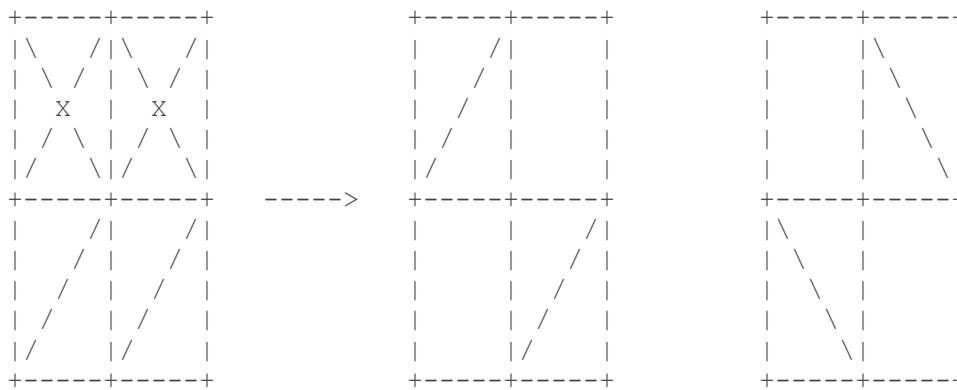
26 2

a kitchen nightmare

**Giải thích:**

**Bộ dữ liệu 1.** Đây là ví dụ của sơ đồ ở trên.

**Bộ dữ liệu 2.** Lưu ý rằng  $N = 2$  và  $K = 1$ , có nghĩa là cấu trúc được cho trong bộ dữ liệu được lặp lại  $N/K = 2$  lần. Có hai cách làm sơ đồ này thành “inspection-friendly” với chi phí nhỏ nhất là **26**, biểu diễn như sau:



**Bộ dữ liệu 3.** Mặc cho đường trung chuyển nào bạn xây, điều đó là không thể để làm sơ đồ này trở thành “inspection-friendly”.