

BÀI TẬP LẬP TRÌNH

BẢNG XOẢN ỐC

Cho bảng vuông kích thước vô hạn, các dòng và cột được thứ tự bắt đầu từ 1. Người ta điền vào bảng các số nguyên từ 1 trở đi và bắt đầu từ ô ở góc trên trái của bảng theo quy luật như hình minh họa với 5 lớp như sau.

Yêu cầu: Cho 2 số nguyên x, y . Hãy tìm giá trị của phần tử tại dòng x , cột y của bảng.

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 9 | 10 | 25 |
| 4 | 3 | 8 | 11 | 24 |
| 5 | 6 | 7 | 12 | 23 |
| 16 | 15 | 14 | 13 | 22 |
| 17 | 18 | 19 | 20 | 21 |

Dữ liệu: Vào từ tập tin **NUMSPIRAL.INP**

- Dòng đầu tiên chứa số nguyên $t (1 \leq t \leq 10^5)$ – số bộ test.
- Mỗi dòng trong t dòng tiếp theo chứa 2 số nguyên $x, y (1 \leq x, y \leq 10^9)$.

Kết quả: Ghi ra tập tin văn bản **NUMSPIRAL.OUT** gồm t dòng, mỗi dòng ghi một số nguyên là giá trị của phần tử tại ô (x, y) của bảng.

Ví dụ:

| NUMSPIRAL . INP | NUMSPIRAL . OUT |
|-----------------|-----------------|
| 3 | 8 |
| 2 3 | 1 |
| 1 1 | 15 |
| 4 2 | |

SƠN NHÀ

Có n căn nhà cần sơn. Căn nhà i được sơn bằng một trong 3 màu Xanh, Hồng, Vàng với mức giá tương ứng là a_{i1}, a_{i2}, a_{i3} .

Yêu cầu: Tìm cách sơn màu cho n ngôi nhà sao cho hai căn nhà cạnh nhau không được sơn cùng màu và tổng chi phí sơn là ít nhất.

Dữ liệu: Vào từ tập tin văn bản **PAINT.INP** chứa số nguyên T là số bộ test ($T \leq 100$). Mỗi bộ test có dạng

- Dòng đầu chứa số nguyên dương n ($1 \leq n \leq 20$) là số ngôi nhà,
- Dòng thứ j trong n dòng sau chứa 3 số nguyên không âm a_{j1}, a_{j2}, a_{j3} là chi phí sơn ngôi nhà j bằng các màu Xanh, Hồng, Vàng tương ứng ($0 \leq a_{jk} \leq 1000$).

Kết quả: Ghi ra tập tin văn bản **PAINT.OUT** gồm T dòng, dòng thứ i là kết quả của bộ test thứ i tương ứng trong dữ liệu vào.

Ví dụ:

| PAINT . INP | PAINT . OUT |
|-------------|-------------|
| 2 | 137 |
| 4 | 96 |
| 13 23 12 | |
| 77 36 64 | |
| 44 89 76 | |
| 31 78 45 | |
| 3 | |
| 26 40 83 | |
| 49 60 57 | |
| 13 89 99 | |

Giải thích test 2: ngôi nhà 1 và 3 sơn màu Xanh, ngôi nhà 2 sơn màu Vàng.

Giới hạn

- 30% số test đầu tiên có $n \leq 3$
- 30% số test tiếp theo có $3 < n \leq 15, T = 1$
- 40% số test cuối có $n \leq 20, T \leq 100$.

CẶP ƯỚC CHUNG LỚN NHẤT

Cho 2 dãy số nguyên dương $A = a_1, a_2, \dots, a_n$ và $B = b_1, b_2, \dots, b_n$. Hãy chọn cặp phân tử $a_i \in A$ và $b_j \in B$ sao cho ước chung lớn nhất của (a_i, b_j) là lớn nhất trong tất cả các cách chọn. Nếu tồn tại nhiều cặp phân tử (a_i, b_j) thỏa yêu cầu thì chọn cặp phân tử có tổng lớn nhất.

Dữ liệu: Vào từ tập tin văn bản **MAXGCD.INP**

- Dòng đầu tiên chứa số nguyên dương $n (n \leq 5 \times 10^5)$
- Dòng thứ hai chứa dãy số nguyên dương $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^6)$
- Dòng thứ ba chứa dãy số nguyên dương $b_1, b_2, \dots, b_n (1 \leq b_j \leq 10^6)$

Kết quả: Ghi ra tập tin văn bản **MAXGCD.OUT** số nguyên dương là tổng lớn nhất tìm được.

Ví dụ:

| MAXGCD . INP | MAXGCD . OUT |
|----------------------------------|--------------|
| 5 7 16 2 11 13 4 1 15 24 9 | 40 |

Giải thích: cặp phân tử 16 và 24 có $\text{UCLN}(16, 24) = 8$ là lớn nhất trong tất cả cặp số.

PHÉP CHIA LẤY PHẦN DƯ

Bờm đã được học về phép toán MOD là phép chia lấy phần dư của 2 số nguyên. Lần này thầy giáo cho bài tập về nhà yêu cầu Bờm đếm số giá trị b nằm trong đoạn từ L đến R thỏa $a \text{ MOD } b = c$.

Hãy giúp Bờm giải bài toán trên.

Dữ liệu: Vào từ tập tin văn bản **MOD.INP** chứa 4 số nguyên a, c, L, R ($0 \leq a, c, L, R \leq 10^9$)

Kết quả: Ghi ra tập tin văn bản **MOD.OUT** một số nguyên dương là kết quả bài toán.

Ví dụ:

| MOD . INP | MOD . OUT |
|-----------|-----------|
| 21 5 1 21 | 2 |

ĐOẠN GỐI

Cho n đoạn thẳng với các điểm đầu a_i và điểm cuối b_i là những số nguyên trong khoảng -1000 đến 1000, $a_i < b_i$. Hãy tìm tối đa k đoạn thẳng gối nhau liên tiếp. Hai đoạn thẳng $[a, b]$ và $[c, d]$ được gọi là gối nhau nếu xếp chúng trên cùng một trục số thì điểm đầu đoạn này trùng với điểm cuối của đoạn kia, nghĩa là $b = c$ hoặc $d = a$.

Dữ liệu: Vào từ tập tin văn bản **OVERLAP.INP**

- Dòng đầu tiên chứa số nguyên dương n ($n \leq 1000$)
- Dòng thứ i trong n dòng tiếp theo, mỗi dòng chứa số nguyên a_i, b_i biểu diễn điểm đầu và điểm cuối của đoạn thứ i

Kết quả: Ghi ra tập tin văn bản **OVERLAP.OUT**

- Dòng đầu ghi số nguyên k
- k dòng tiếp theo, mỗi dòng ghi chỉ số của các đoạn thẳng gối nhau liên tiếp

Ví dụ:

| OVERLAP . INP | OVERLAP . OUT |
|---------------|---------------|
| 5 | 3 |
| 2 7 | 2 |
| 1 3 | 4 |
| 7 9 | 5 |
| 3 4 | |
| 4 5 | |

TỔNG CHẴN LỚN NHẤT

Cho lưới hình chữ nhật gồm m hàng, các hàng đánh số từ 1 đến m và n cột, các cột đánh số từ 1 đến n , giao của hàng i cột j chứa số nguyên dương a_{ij} . Từ ô (i, j) chỉ có thể di chuyển đến các ô $(i, j + 1)$ và $(i + 1, j)$. Giá trị của lưới là tổng giá trị các ô là số chẵn thuộc một đường đi từ ô $(1, 1)$ đến ô (m, n) .

Yêu cầu: Xác định giá trị lớn nhất của lưới.

Dữ liệu: Vào từ tập tin văn bản **MAXSUM.INP**

- Dòng đầu chứa 2 số m và n ($m, n \leq 1000$),
- Mỗi dòng trong m dòng tiếp theo chứa n số nguyên dương không vượt quá 10^9 mô tả dữ liệu của lưới hình chữ nhật.

Kết quả: Ghi ra tập tin văn bản **MAXSUM.OUT** số nguyên là theo yêu cầu của bài toán.

Ví dụ:

| MAXSUM . INP | MAXSUM . OUT |
|-----------------------|--------------|
| 2 3 2 3 4 3 2 2 | 8 |

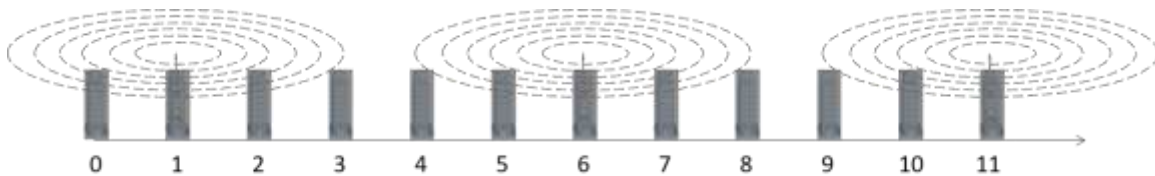
TRẠM PHÁT SÓNG

Thành phố Wonderland có n khu dân cư được quy hoạch cách đều nhau một đơn vị độ dài dọc theo một con đường thẳng tắp. Như vậy thành phố có thể được biểu diễn bởi trục số và vị trí n khu dân cư là các điểm có tọa độ nguyên liên tiếp từ 0 đến $n - 1$.

Chính quyền thành phố lên kế hoạch xây dựng mới các trạm phát sóng di động sao cho mọi khu dân cư đều nhận được sóng. Mỗi trạm phát phủ sóng một vùng có bán kính R với tâm là trạm phát sóng.

Có m vị trí trên trục đường có thể xây dựng trạm phát sóng, vị trí thứ i có tọa độ x_i . Nhà quản lý muốn chọn một số vị trí lắp đặt trạm sao cho mọi khu dân cư đều nhận được sóng.

Chẳng hạn $n = 12$, có thể xây dựng trạm tại 8 vị trí với tọa độ là: 3, 1, 0, 5, 9, 6, 10, 11. Với bán kính phát sóng $R = 2$, ta chỉ cần xây 3 trạm tại các tọa độ 1, 6, 11, tương ứng tại các vị trí 2, 6 và 8 thì có thể phủ sóng cho tất cả n khu dân cư.



Yêu cầu: Hãy tìm phương án chọn ra ít nhất các vị trí cần xây dựng trạm sao cho toàn bộ n khu dân cư đều nhận sóng. Nếu không tìm được phương án thì ghi 0.

Dữ liệu: Vào từ tập tin văn bản **STATION.INP** trong đó:

- Dòng đầu ghi 3 số n, m và R ($1 \leq n \leq 10^6$; $1 \leq m \leq n$; $1 \leq R \leq 10^6$)
- Dòng thứ hai ghi m số x_1, x_2, \dots, x_m ($0 \leq x_i \leq 10^6$)

Kết quả: Ghi ra tập tin văn bản **STATION.OUT** gồm hai dòng:

- Dòng đầu ghi số k ($k \geq 0$) – số trạm cần xây dựng,
- Nếu $k > 0$ thì dòng thứ hai ghi ra k chỉ số các vị trí cần xây dựng trạm.

Ví dụ:

| STATION . INP | STATION . OUT |
|-----------------------------|---------------|
| 12 8 2 3 1 0 5 9 6 10 11 | 3 2 6 8 |

NHÀ MÁY ĐIỆN Ở GOODLAND

Goodland là quốc gia có n thành phố, thành phố c_i có số thứ tự từ 0 đến $n - 1$. Các thành phố được kết nối với nhau bởi $n - 1$ con đường, mỗi con đường kết nối thành phố c_i và c_{i+1} . Khoảng cách giữa thành phố c_i và c_j là $|i - j|$.

Chính phủ Goodland triển khai một dự án cải tạo cơ sở hạ tầng của quốc gia và mang điện đến cư dân của đất nước. Mỗi một thành phố được phép xây dựng tối đa một nhà máy phát điện nhưng chúng đều chưa đi vào hoạt động. Khi hoạt động, mỗi nhà máy sản xuất điện đủ cung cấp cho tất cả thành phố lân cận có khoảng cách $< k$ tính từ nhà máy.

Yêu cầu: Hãy giúp chính phủ tìm số nhà máy điện ít nhất cần cấp phép hoạt động để đảm bảo tất cả thành phố của Goodland đều có điện.

Dữ liệu: Vào từ tập tin văn bản **ELECTRICITY.INP**

- Dòng đầu chứa hai số nguyên $n, k (1 \leq k \leq n \leq 10^5)$.
- Dòng thứ hai chứa dãy gồm n số, số thứ i có giá trị 0 hoặc 1 tương ứng với số nhà máy điện xây dựng tại thành phố c_i .

Kết quả: Ghi ra tập tin văn bản **ELECTRICITY.OUT** một số nguyên là số nhà máy điện ít nhất cần cấp phép hoạt động. Nếu không tìm được lời giải thì thông báo NO SOLUTION.

Ví dụ:

| ELECTRICITY . INP | ELECTRICITY . OUT |
|--------------------------|--------------------------|
| 6 2 0 1 1 1 1 0 | 2 |

THAM QUAN

Điều kiện làm việc cho phép trong khoảng thời gian n ngày liên tục nhân viên có thể đăng ký nghỉ phép hoặc nghỉ bù. Steve quyết định tận dụng khả năng hiếm có này để nghỉ ngơi liên tục tất cả k ngày mà mình được quyền.

Sau khi cân nhắc, Steve đăng ký tham gia một tour du lịch. Mỗi ngày công ty du lịch sẽ đưa khách tới tham quan một thành phố. Trong bảng giới thiệu chương trình tham quan mỗi thành phố được ghi tương ứng với một số nguyên dương không vượt quá 10^9 , ngày thứ i sẽ tham thành phố a_i . Các thành phố khác nhau tương ứng với các số nguyên khác nhau. Một thành phố có thể được tới nhiều lần. Steve có thể tham gia tour du lịch bắt đầu từ ngày tùy chọn và dĩ nhiên, muốn thăm được càng nhiều thành phố khác nhau càng tốt.

Yêu cầu: Hãy xác định ngày bắt đầu kỳ nghỉ để nghỉ đủ k ngày và số lượng các thành phố Steve có thể tới thăm là lớn nhất. Ngày đầu tiên trong chương trình của Công ty được đánh số là 1. Nếu tồn tại nhiều cách lựa chọn thì đưa kết quả bất kỳ.

Dữ liệu: Vào từ tập tin văn bản **EXCURS.INP**

- Dòng thứ nhất chứa 2 số nguyên $n, k (1 \leq k \leq n \leq 10^5)$.
- Dòng thứ hai chứa n số nguyên $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^9)$.

Kết quả: Ghi ra tập tin văn bản **EXCURS.OUT** một số nguyên – ngày được chọn để bắt đầu tham quan.

Ví dụ:

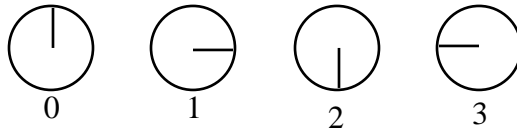
| EXCURS . INP | EXCURS . OUT |
|----------------------|--------------|
| 7 3 1 2 1 3 1 2 1 | 2 |

CLOCKS

Cho 9 chiếc đồng hồ được sắp thành ma trận 3×3 như sau:

| | | |
|---|---|---|
| A | B | C |
| D | E | F |
| G | H | I |

Mỗi đồng hồ ở 1 trong 4 trạng thái sau:



Cho 9 phép biến đổi, mỗi biến đổi tác động lên một số đồng hồ nhất định và quay các kim một góc 90° .

| Phép biến đổi | Các đồng hồ bị tác động |
|---------------|-------------------------|
| 1 | A, B, D, E |
| 2 | A, B, C |
| 3 | B, C, E, F |
| 4 | A, D, G |
| 5 | B, D, E, F, H |
| 6 | C, F, I |
| 7 | D, E, G, H |
| 8 | G, H, I |
| 9 | E, F, H, I |

Yêu cầu: Cho trước trạng thái ban đầu của các đồng hồ. Hãy thực hiện một số ít nhất các phép biến đổi trên để đưa tất cả đồng hồ về trạng thái 0

Dữ liệu: Vào từ tập tin văn bản **CLOCKS.INP** gồm 3 dòng, mỗi dòng gồm 3 số nguyên có giá trị từ 0 đến 3 mô tả trạng thái ban đầu của các đồng hồ được sắp thành ma trận 3×3

Kết quả: Ghi ra tập tin văn bản **CLOCKS.OUT**

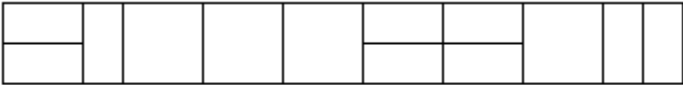
- Dòng đầu tiên ghi số phép biến đổi tối thiểu
- Các dòng tiếp theo ghi chỉ số các phép biến đổi cần thực hiện theo thứ tự bất kỳ, mỗi số trên một dòng
- Trường hợp không tìm được lời giải thì ghi NO SOLUTION

Ví dụ:

| CLOCKS . INP | CLOCKS . OUT |
|--------------|--------------|
| 0 2 3 | 3 |
| 2 1 2 | 5 |
| 3 2 0 | 7 |

LÁT VIỀN

Đường viền trang trí ở nền nhà có kích thước $2 \times n$ được lát bằng 2 loại gạch: loại kích thước 1×2 và loại 2×2 . Hãy xác định số cách lát khác nhau có thể thực hiện.



Dữ liệu: Vào từ tập tin văn bản **TILING.INP** gồm nhiều dòng, mỗi dòng chứa một số nguyên dương n ($n \leq 50$).

Kết quả: Ghi ra tập tin văn bản **TILING.OUT** các kết quả tìm được, mỗi số trên một dòng.

Ví dụ:

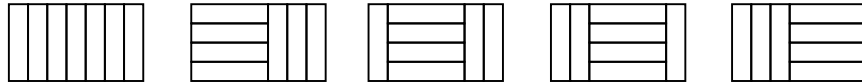
| TILING . INP | TILING . OUT |
|---------------------|---------------------|
| 2 | 3 |
| 8 | 171 |
| 12 | 2731 |

LÁT SÂN NHÀ

Bờm và Phú Ông đang ngồi trước sân tận hưởng những ngọn gió mát của buổi chiều hè. Cả hai cảm thấy buồn chán và Phú Ông nảy ra một trò chơi và treo thưởng nếu Bờm giải được sẽ thưởng cho một gói xôi, món ăn yêu thích của Bờm.

Sân nhà Phú Ông là một hình chữ nhật kích thước $4 \times n$ ô vuông đơn vị. Nhà Phú Ông giàu có cũng đang chất một đồng gạch loại 1×4 với số lượng không biết bao nhiêu mà kể. Phú Ông yêu cầu Bờm hãy cho biết số cách khác nhau để lát kín mặt sân bằng những viên gạch đó. Tuy nhiên, Phú Ông sợ Bờm có thể dễ dàng giải câu đố của mình nên đã yêu cầu Bờm đếm số lượng số nguyên tố có giá trị không vượt quá số cách lát mặt sân.

Chẳng hạn với $n = 7$, ta có 5 cách lát khác nhau như sau



Khi đó có 3 số nguyên tố không vượt quá 5 là 2,3,5.

Yêu cầu: Cho số nguyên $n (1 \leq n \leq 40)$. Hãy giúp Bờm đếm số lượng số nguyên tố thỏa yêu cầu của Phú Ông.

Dữ liệu: Vào từ tập tin văn bản **YARD.INP**

- Dòng đầu tiên chứa số nguyên dương $T (1 \leq T \leq 20)$ – số câu đố của Phú Ông.
- Mỗi dòng trong T dòng tiếp theo chứa số nguyên $n (1 \leq n \leq 40)$.

Kết quả: Ghi ra tập tin văn bản **YARD.OUT** gồm T dòng, mỗi dòng là kết quả của câu đố tương ứng.

Ví dụ:

| YARD . INP | YARD . OUT |
|------------|------------|
| 1 7 | 3 |

DÒ TÌM MẬT KHẨU

Phòng thí nghiệm của trung tâm công nghệ cao được trang bị khá nhiều thiết bị máy móc hiện đại phục vụ cho công việc nghiên cứu. Các cửa ra vào đều được trang bị hệ thống nhận dạng thẻ từ hiện đại. Tuy nhiên, các nhà nghiên cứu thường là những người đăng trí nên họ hay dễ quên hoặc làm rớt thẻ từ.

Mặc dù các nhà nghiên cứu hay đăng trí nhưng họ rất thông minh và giỏi lập trình nên giám đốc liền nghĩ ra một cách thức nhận dạng mới mà không cần dùng đến thẻ từ. Trước mỗi cửa ra vào, giám đốc cho thiết kế một bảng điện tử cứ sau mỗi giây sẽ hiển thị 1 dãy hàng ngang gồm rất nhiều số nguyên dương. Mật khẩu để mở cửa chính là chuỗi số được tạo thành bằng cách ghép cặp số ở 2 vị trí khác nhau theo đúng thứ tự xuất hiện của chúng trong dãy thỏa điều kiện tổng các chữ số của cặp số này là khác nhau và lệch nhau nhỏ nhất. Nếu trong dãy tồn tại nhiều cặp phân tử như thế thì chọn cặp phân tử trái nhất.

Ví dụ dãy số hiển thị trên bảng như sau:

| | | | | | | | |
|-------------------------|----|-----|----|-----|----|------|----|
| Dãy số | 75 | 305 | 19 | 221 | 43 | 1592 | 36 |
| Giá trị tổng các chữ số | 12 | 8 | 10 | 5 | 7 | 20 | 9 |

Cặp số 305 và 43 là cặp số đầu tiên tính từ trái sang có tổng các chữ số là 8 và 7 tương ứng, độ lệch tổng các chữ số của cặp này là 1, trong trường hợp này mật khẩu để mở cửa là 30543. Một nghiên cứu viên tập sự đang gặp khó khăn trong việc xác định mật khẩu theo phương pháp của giám đốc đưa ra. Bạn hãy giúp nghiên cứu viên này tìm ra mật khẩu để mở cửa trong thời gian nhanh nhất có thể.

Dữ liệu: Vào từ tập tin văn bản **PASSWORD.INP**:

- Dòng đầu ghi số nguyên dương $n (n \leq 10^6)$
- Dòng tiếp theo ghi n số nguyên $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^9)$ là các số hiển thị trên bảng điện tử
- Dữ liệu được cho đảm bảo tồn tại ít nhất 2 phân tử có tổng các chữ số là khác nhau

Kết quả: Ghi ra tập tin văn bản **PASSWORD.OUT** là mật khẩu tìm được.

Ví dụ:

| PASSWORD . INP | PASSWORD . OUT |
|-------------------------------|----------------|
| 7 75 305 19 221 43 1592 36 | 30543 |

SINH XÂU NHỊ PHÂN

Cho 1 một xâu nhị phân S độ dài n . Tìm xâu nhị phân kế tiếp của xâu S trong thứ tự từ điển.

Dữ liệu: Vào từ tập tin văn bản **BINARYGEN.INP**

- Dòng đầu chứa số nguyên dương $n (n \leq 10^4)$
- Dòng tiếp theo ghi n số 0 hoặc 1 liên tiếp nhau.

Kết quả: Ghi ra tập tin văn bản **BINARYGEN.OUT** xâu nhị phân kế tiếp của xâu S trên một dòng duy nhất. Nếu không tồn tại thì ghi ra -1.

Ví dụ:

| BINARYGEN . INP | BINARYGEN . OUT |
|-----------------|-----------------|
| 5 00100 | 00101 |

SINH CHUỖI TỔ HỢP

Cho dãy số nguyên C độ dài m là một tổ hợp chập m phần tử của tập $S = \{1, 2, \dots, n\}$. Tìm dãy tổ hợp kế tiếp của C trong thứ tự từ điển.

Dữ liệu: Vào từ tập tin văn bản **COMBINATIONGEN.INP**

- Dòng đầu chứa 2 số nguyên dương $n, m (n, m \leq 10^4)$
- Dòng tiếp theo ghi m số nguyên dương thuộc tập S .

Kết quả: Ghi ra tập tin văn bản **COMBINATIONGEN.OUT** dãy tổ hợp kế tiếp của C . Nếu không tồn tại thì ghi ra -1.

Ví dụ:

| COMBINATIONGEN . INP | COMBINATIONGEN . OUT |
|----------------------|----------------------|
| 5 3 2 3 5 | 2 4 5 |

SINH HOÁN VỊ

Cho dãy H độ dài n là một hoán vị của tập $S = \{1, 2, \dots, n\}$. Tìm hoán vị kế tiếp của H trong thứ tự từ điển.

Dữ liệu: Vào từ tập tin văn bản **PERMUTATIONGEN.INP**

- Dòng đầu chứa số nguyên dương $n (n \leq 10^4)$
- Dòng tiếp theo ghi dãy gồm n số nguyên dương là các phần tử của H .

Kết quả: Ghi ra tập tin văn bản **PERMUTATIONGEN.OUT** dãy số nguyên là hoán vị kế tiếp của H . Nếu không tồn tại thì ghi ra -1 .

Ví dụ:

| PERMUTATIONGEN . INP | PERMUTATIONGEN . OUT |
|----------------------|----------------------|
| 5 3 2 1 5 4 | 3 2 4 1 5 |

SINH CHUỖI

Cho chuỗi kí tự độ dài không quá 8 chỉ gồm các kí tự Latin in thường. Hãy sinh ra tất cả chuỗi khác nhau được tạo thành bởi các kí tự của chuỗi ban đầu.

Dữ liệu: Vào từ tập tin văn bản **GENSTR.INP** chứa chuỗi độ dài không quá 8 chỉ gồm các kí tự 'a', 'b', ..., 'z'.

Kết quả: Ghi ra tập tin văn bản **GENSTR.OUT**

- Dòng đầu tiên chứa số nguyên n – số chuỗi khác nhau sinh được.
- Các chuỗi sinh ra được in trên các dòng khác nhau tiếp theo và in theo thứ tự từ điển.

Ví dụ:

| GENSTR . INP | GENSTR . OUT |
|--------------|---|
| aabac | 20 aaabc aaacb aabac aabca aacab aacba abaac abaca abcaa acaab acaba acbaa baaac baaca baca bcaaa caaab caaba cabaa cbaaa |

ĐỔI TIỀN

Cho m loại tiền với số lượng vô hạn, loại thứ i có giá trị c_i đồng. Hãy đếm số cách lấy n đồng từ các loại tiền đã cho.

Dữ liệu: Vào từ tập tin văn bản **COINCHANGE.INP**

- Dòng đầu tiên chứa 2 số nguyên n, m ($1 \leq n \leq 250; 1 \leq m \leq 50$).
- Dòng thứ hai chứa dãy c_1, c_2, \dots, c_m ($1 \leq c_i \leq 50$) – giá trị các loại tiền tương ứng. Dữ liệu đảm bảo giá trị các loại tiền là phân biệt.

Kết quả: Ghi ra tập tin văn bản **COINCHANGE.OUT** một số nguyên là số cách lấy ra n đồng từ các loại tiền.

Ví dụ:

| COINCHANGE . INP | COINCHANGE . OUT |
|------------------|------------------|
| 10 4 2 5 3 6 | 5 |

BỜM ĐI HỘI CHỢ

Bờm được mẹ cho một số tiền M để đi chơi hội chợ. Sau khi dạo qua các gian hàng, Bờm nhận thấy có n món đồ mà cậu ta rất thích lần lượt có giá tiền c_1, c_2, \dots, c_n . Các món đồ đều có số lượng vô cùng phong phú. Bờm muốn dùng toàn bộ số tiền mẹ cho để mua những món đồ này, mỗi món có thể mua với số lượng nhiều hơn 1 để dành tặng cho các bạn trong lớp. Bờm cũng không thích mang nhiều tiền dư khi rời hội chợ nên muốn xài tiền nhiều nhất có thể.

Yêu cầu: Cho số nguyên M và dãy gồm n số nguyên c_1, c_2, \dots, c_n . Hãy tính số tiền nhiều nhất mà Bờm có thể dùng để mua các món hàng yêu thích trong khu hội chợ.

Dữ liệu: Vào từ tập tin văn bản **EXPENSE.INP**

- Dòng đầu chứa số nguyên $t (t \leq 10)$ là số bộ test.
- Mỗi bộ test gồm 2 dòng
 - + Dòng đầu chứa 2 số nguyên $n, M (1 \leq n, M \leq 2000)$
 - + Dòng tiếp theo chứa dãy $c_1, c_2, \dots, c_n (1 \leq c_i \leq 2000)$

Kết quả: Ghi ra tập tin văn bản **EXPENSE.OUT** gồm t dòng, mỗi dòng là kết quả cho mỗi bộ test tương ứng.

Ví dụ:

| EXPENSE . INP | EXPENSE . OUT |
|---------------|---------------|
| 2 | 13 |
| 4 13 | 10 |
| 3 7 9 11 | |
| 3 11 | |
| 3 7 9 | |

KHÔNG ĐƠN GIẢN

Không phải cứ ngắn gọn là dễ! Bạn có thể kiểm chứng điều đó ở bài toán sau: Cho 3 số nguyên a, b, c ($1 \leq a, b, c \leq 10^{19}$). Hãy tìm số dư khi chia a^b cho c .

Dữ liệu: Vào từ tập tin văn bản **NOSIMPLE.INP** gồm một dòng chứa 3 số nguyên a, b, c .

Kết quả: Ghi ra tập tin văn bản **NOSIMPLE.OUT** một số nguyên – số dư tìm được.

Ví dụ:

| NOSIMPLE . INP | NOSIMPLE . OUT |
|----------------|----------------|
| 3 4 5 | 1 |

TÌM CHỮ SỐ

Xét biểu diễn thập phân của phân số a/b . Biểu diễn này có thể là một số thập phân hữu hạn hoặc một số thập phân vô hạn tuần hoàn. Nếu phân số có thể biểu diễn bởi một số thập phân hữu hạn, ta có thể viết thêm một dãy vô hạn các chữ số 0 vào sau chữ số cuối cùng sau dấu chấm thập phân và coi đó cũng là một số thập phân vô hạn tuần hoàn. Ví dụ:

$$\begin{aligned}\frac{100}{8} &= 12,500 \dots 0 \dots \\ \frac{17}{3} &= 5,66 \dots 6 \dots \\ \frac{99}{140} &= 0,70714285714285 \dots 714285 \dots\end{aligned}$$

Yêu cầu: Các chữ số đứng sau dấu “,” trong biểu diễn thập phân của a/b sau đó được đánh số từ 1 trở đi, từ trái qua phải. Hãy xác định chữ số đứng thứ k .

Dữ liệu: Vào từ tập tin văn bản **DIGIT.INP** gồm 1 dòng chứa ba số nguyên dương $a, b, k < 10^{18}$ cách nhau ít nhất một dấu cách.

Kết quả: Ghi ra tập tin văn bản **DIGIT.OUT** một số nguyên duy nhất là giá trị chữ số tìm được

Ví dụ:

| DIGIT . INP | DIGIT . OUT |
|-------------|-------------|
| 17 3 10 | 6 |
| | |

PHỐ MAY MẮN

Người dân thành phố Byteland có rất nhiều điều kiêng kị trong cuộc sống. Theo quan điểm của họ, các số 2, 6, 13 và nhiều số khác không mang lại điều may mắn. Trong khi đó các số 3, 5, 7 lại rất được ưa chuộng. Những ngôi nhà có số mà khi phân tích ra thừa số nguyên tố chỉ chứa các thừa số 3, 5, 7 được coi là may mắn và được mua rất nhanh.

Sau một thời gian dài thảo luận, hội đồng thành phố quyết định đánh số tất cả ngôi nhà trên một đường phố mới xây bằng các số may mắn liên tiếp nhau, biến phố đó thành một phố may mắn. Kí hiệu dãy các số may mắn là X_1, X_2, X_3, \dots . Khi đó, các nhà bên trái mang số X_1, X_3, X_5, \dots còn các nhà bên phải mang số X_2, X_4, X_6, \dots . Toàn bộ đường phố không có quá 1000 ngôi nhà.

Yêu cầu: Hãy xác định xem số cho trước có phải là một số nhà ở phố may mắn hay không, nếu đúng thì cho biết nhà đó nằm bên trái hay bên phải của phố.

Dữ liệu: Vào từ tập tin văn bản **STREET.INP** gồm không quá $t (t \leq 10^5)$ dòng, mỗi dòng chứa 1 số nguyên dương.

Kết quả: Ghi ra tập tin văn bản **STREET.OUT** gồm nhiều dòng, mỗi dòng tương ứng với 1 số ở tập tin dữ liệu vào và chứa 1 trong 3 chữ cái L, R, N phụ thuộc vào số đó tương ứng với nhà bên trái, bên phải hay không phải là số nhà.

Ví dụ:

| STREET . INP | STREET . OUT |
|--------------|--------------|
| 5 | R |
| 3 | L |
| 4 | N |

XÂU KÝ TỰ NGOẶC

Xét xâu chỉ chứa các ký tự ngoặc tròn (,), ngoặc vuông [,] và ngoặc nhọn {, }. Để ngắn gọn, ta gọi nó là xâu ngoặc.

Định nghĩa xâu ngoặc đúng:

- Xâu rỗng được coi là xâu ngoặc đúng,
- Nếu a là xâu ngoặc đúng thì (a) , $[a]$, $\{a\}$ cũng là các xâu ngoặc đúng,
- Nếu a và b là các xâu ngoặc đúng thì ab cũng là xâu ngoặc đúng.

Cho xâu S độ dài n . Xâu $s_k s_{k+1} s_{k+2} \dots s_n s_1 s_2 \dots s_{k-1}$ được gọi là xâu đầy vòng của S . Bản thân S cũng là một xâu đầy vòng của S .

Yêu cầu: Cho xâu ngoặc S có độ dài không quá 1000. Hãy xác định có tồn tại một xâu đầy vòng của S là xâu ngoặc đúng hay không và đưa ra câu trả lời Yes hoặc No.

Dữ liệu: Vào từ tập tin văn bản **BRACKETS.INP** gồm một dòng chứa xâu S .

Kết quả: Ghi ra tập tin văn bản **BRACKETS.OUT** câu trả lời Yes hoặc No.

Ví dụ:

| BRACKETS . INP | BRACKETS . OUT |
|----------------|----------------|
| } { } () { | Yes |

PHÂN MÁY MÁY GIA CÔNG

Một xưởng máy có 2 máy gia công sản phẩm, máy thứ nhất có hiệu suất làm việc nhanh gấp đôi máy thứ hai. Xưởng máy nhận đơn đặt hàng gồm n sản phẩm, sản phẩm thứ i có thời gian gia công t_i . Tại mỗi thời điểm, mỗi máy chỉ gia công 1 sản phẩm. Sau khi hoàn thiện một sản phẩm, máy sẽ được phân phối để gia công sản phẩm tiếp theo.

Thợ xưởng phân phối n sản phẩm cho 2 máy theo cách sau:

- Máy thứ nhất gia công bắt đầu từ sản phẩm thứ 1 trở đi (thứ tự thực hiện: 1, 2, ...).
- Máy thứ hai gia công bắt đầu từ sản phẩm thứ n trở lại (thứ tự thực hiện: $n, n - 1, \dots$).
- Khi chỉ còn 1 sản phẩm chưa gia công mà cả 2 máy đều rồi thì phân cho máy thứ nhất thực hiện.

Yêu cầu: Cho biết số lượng sản phẩm mà mỗi máy gia công được theo cách phân phối trên.

Dữ liệu: Vào từ tập tin văn bản **DISTRIBUTION.INP**

- Dòng đầu tiên chứa số nguyên $t (t \leq 100)$ – số bộ test.
- Dòng đầu tiên của mỗi bộ test chứa số nguyên $n (1 \leq n \leq 10^5)$ – số sản phẩm.
- Dòng thứ hai của mỗi bộ test chứa dãy số nguyên $t_1, t_2, \dots, t_n (1 \leq t_i \leq 10^9)$ – thời gian để gia công các sản phẩm tương ứng.

Kết quả: Ghi ra tập tin văn bản **DISTRIBUTION.OUT** gồm t dòng, mỗi dòng ghi 2 số nguyên x, y – tương ứng số lượng sản phẩm mà máy thứ nhất và máy thứ hai gia công được.

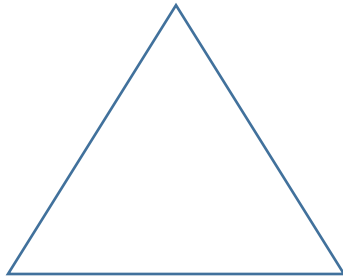
Ví dụ:

| DISTRIBUTION . INP | DISTRIBUTION . OUT |
|--------------------|--------------------|
| 3 | 4 1 |
| 5 | 1 3 |
| 2 6 2 1 7 | 3 2 |
| 4 | |
| 15 2 1 3 | |
| 5 | |
| 2 4 12 4 7 | |

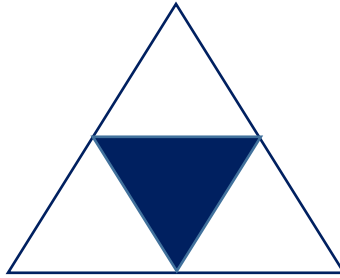
ĐẾM HÌNH TAM GIÁC

Các bài tập tin học gần đây khiến Bờm đau đầu vì toàn những bài khó. Trong lúc vẽ vờn nguệch ngoạc vào giấy nháp vì chưa tìm ra lời giải tối ưu, Bờm phát hiện những điều thú vị qua những hình vẽ của mình.

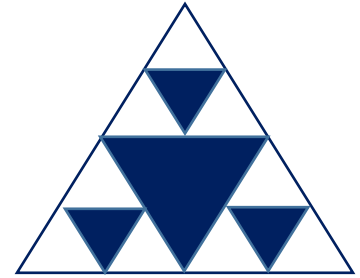
Ban đầu Bờm vẽ một tam giác đều. Ở lần vẽ thứ $i (i = 1, 2, \dots)$, Bờm chia mỗi tam giác đều không bị tô màu thành 4 tam giác đều nhỏ hơn bên trong và tô màu tam giác nằm ở vị trí trung tâm như hình minh họa.



Ban đầu



Sau lần 1 (có 5 tam giác)



Sau lần 2 (có 17 tam giác)

Yêu cầu: Cho số nguyên dương n . Hãy đếm số lượng hình tam giác nhìn thấy được sau lần vẽ thứ n .

Dữ liệu: Vào từ tập tin văn bản **NUMTRIS.INP**

- Dòng đầu tiên chứa số nguyên dương $t (1 \leq t \leq 2 \times 10^5)$ – số bộ test.
- Mỗi dòng trong t dòng tiếp theo chứa số nguyên dương $n (n \leq 10^{18})$.

Kết quả: Ghi ra tập tin văn bản **NUMTRIS.OUT** gồm t dòng, mỗi dòng là phần dư của kết quả tương ứng khi chia cho $10^9 + 7$.

Ví dụ:

| NUMTRIS . INP | NUMTRIS . OUT |
|---------------|---------------|
| 3 | 5 |
| 1 | 17 |
| 2 | 53 |
| 3 | |

VIẾT BÁO CÁO

Tí đang chuẩn bị viết một báo cáo về trận thi đấu thể thao vừa kết thúc. Cậu đã có đầy đủ ý tưởng trong đầu, việc còn lại là viết nó ra. Bản báo cáo gồm có 2 phần: phần đầu gồm có n từ, từ thứ i có a_i kí tự; phần sau gồm có m từ, từ thứ j có b_j kí tự. Bản báo cáo của Tí không chứa dấu câu.

Cả hai phần của bản báo cáo được viết theo cùng một quy tắc sau:

- Mỗi kí tự chiếm đúng 1 đơn vị độ rộng trên trang giấy.
- Mỗi phần được canh theo lề trái của chúng và bắt đầu từ dòng 1.
- Các từ được viết lần lượt trên cùng dòng với từ liền trước và cách nhau đúng 1 khoảng trắng.
- Khi độ dài của từ lớn hơn độ rộng còn lại của phần trang giấy thì từ này được viết ở dòng tiếp theo và sát lề trái.

Tí muốn kẻ một đường dọc để chia đôi trang giấy có độ rộng w thành 2 phần: phần trái dùng để viết phần đầu của báo cáo, phần phải viết phần sau của báo cáo.

Yêu cầu: Cho độ rộng w của trang giấy và 2 dãy số nguyên a_1, a_2, \dots, a_n và b_1, b_2, \dots, b_m mô tả độ dài của từng từ trong 2 phần tương ứng của bản báo cáo. Hãy tìm cách kẻ một đường dọc chia đôi trang giấy sao cho chiều dài (tính bằng số dòng) của bản báo cáo là nhỏ nhất.

Dữ liệu: Vào từ tập tin văn bản **REPORT.INP**

- Dòng đầu tiên chứa 3 số nguyên w, n, m ($1 \leq w \leq 10^9; 1 \leq n, m \leq 10^5$).
- Dòng thứ hai chứa dãy số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).
- Dòng thứ ba chứa dãy số nguyên b_1, b_2, \dots, b_m ($1 \leq b_j \leq 10^9$).
- Dữ liệu đảm bảo luôn có cách chia đôi trang giấy để viết bản báo cáo.

Kết quả: Ghi ra tập tin văn bản **REPORT.OUT** một số nguyên là số dòng ít nhất của bản báo cáo.

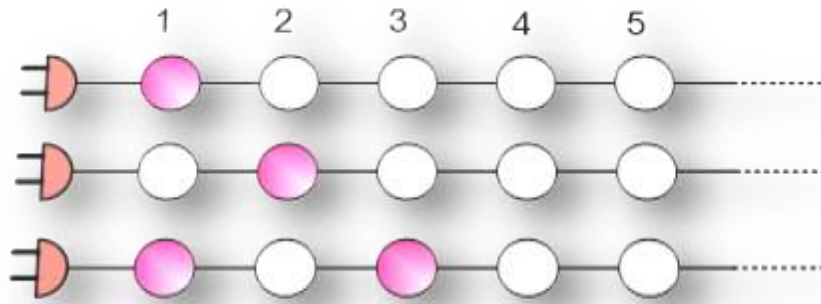
Ví dụ:

| REPORT . INP | REPORT . OUT |
|--------------------------------------|--------------|
| 15 6 6 2 2 2 3 2 2 3 3 5 2 4 3 | 3 |

Giải thích: kẻ đường dọc giữa cột 7 và cột 8. Phần đầu có 3 dòng: (2, 2), (2, 3), (2, 2), phần sau có 3 dòng: (3, 3), (5, 2), (4, 3).

ĐÈN TRANG TRÍ

Đường phố chính của thành phố trong ngày lễ được trang trí bằng một dây đèn nhấp nháy duy nhất dài vô hạn, quấn vào cây và các biểu tượng chào mừng chạy dọc suốt tuyến phố. Đèn nhấp nháy theo quy tắc sau: khi mới cắm điện duy nhất chỉ có đèn sát ô điện sáng. Cứ mỗi giây mỗi bóng đèn kiểm tra, nếu cạnh nó có đúng một bóng đèn sáng thì bóng đèn này cũng sẽ sáng ở giây tiếp theo, trong trường hợp ngược lại – sẽ tắt. Bóng đèn đầu tiên chỉ có một bóng ở cạnh nó.



Yêu cầu: Cho số nguyên $n (1 \leq n \leq 10^{18})$. Hãy xác định xem ở giây thứ n có bao nhiêu bóng đèn sáng.

Dữ liệu: Vào từ tập tin văn bản **LIGHT.INP** chứa số nguyên n .

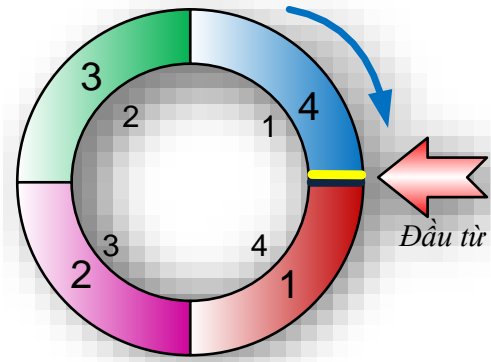
Kết quả: Ghi ra tập tin văn bản **LIGHT.OUT** một số nguyên – số bóng đèn sáng.

Ví dụ:

| LIGHT . INP | LIGHT . OUT |
|-------------|-------------|
| 5 | 2 |

ĐỌC FILE

Đĩa cứng máy tính chứa file gồm n khối đánh số từ 1 đến n theo trình tự tạo ra và cũng là trình tự phải đọc ($1 \leq n \leq 10^5$). Các khối đều nằm trên một rãnh từ, nhưng do đĩa bị phân mảnh nên các khối có thể ở những vị trí khác nhau và không theo thứ tự từ 1 đến n theo trình tự tạo file. Số khối ghi được trên một rãnh từ là n . Đĩa từ luôn quay và theo một chiều cố định. Đầu đọc ở một vị trí cố định. Ban đầu, đầu từ ở đầu của khối thứ nhất của rãnh từ. Cứ mỗi lần đầu từ đi từ khối cuối cùng của rãnh, vượt qua điểm đầu của khối thứ nhất thì đĩa từ được coi là thực hiện trọn vẹn một vòng quay. Vì hiệu ứng phân mảnh nên để đọc hết các khối theo trình tự tạo file có khi phải cần tới vài vòng quay trọn vẹn của đĩa. Ví dụ, với $n = 4$ và các khối có trình tự theo chiều quay là 4, 3, 2, 1 cần tới 3 vòng quay trọn vẹn của đĩa.



Yêu cầu: Cho n và các số p_i – thứ tự của khối trong file ghi ở vị trí thứ i của rãnh ($1 \leq i \leq n$). Hãy xác định số vòng quay trọn vẹn của đĩa cần thực hiện để đọc hết file.

Dữ liệu: Vào từ tập tin văn bản **RFILE.INP**

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ 2 chứa n số nguyên p_1, p_2, \dots, p_n .

Kết quả: Ghi ra tập tin văn bản **RFILE.OUT** một số nguyên – số vòng quay trọn vẹn của đĩa cần thực hiện.

Ví dụ:

| RFILE . INP | RFILE . OUT |
|-------------|-------------|
| 4 | 3 |
| 4 3 2 1 | |

SÁNG TÁC THƠ

Bờm sáng tác một bài thơ có quy tắc gieo vần như sau: nếu 2 câu cách nhau đúng m dòng thì k ký tự cuối của các câu đó phải giống nhau, bao gồm cả dấu cách, giống nhau cả cách viết hoa hay thường.

Bờm đã sáng tác được n dòng thơ. Bờm đề nghị Phú Ông sáng tác tiếp. Bờm có giải thích cho Phú Ông về quy tắc gieo vần nhưng do tuổi cao nên khi Bờm rời đi thì Phú Ông đã quên mất giá trị k .

Không muốn bị mất mặt với Bờm, việc đầu tiên của Phú Ông là dựa vào n dòng thơ đã được sáng tác trước đó để xác định giá trị k lớn nhất thỏa mãn quy tắc gieo vần.

Yêu cầu: Hãy xác định giá trị k lớn nhất mà Phú Ông sử dụng khi kéo dài bài thơ.

Dữ liệu: Vào từ tập tin văn bản **LYRICS.INP**

- Dòng đầu tiên chứa 2 số nguyên n, m ($1 \leq m \leq n \leq 1000$).
- Mỗi dòng trong n dòng sau chứa một dòng thơ dưới dạng xâu độ dài không quá 1000 ký tự chữ cái la tinh (hoa hoặc thường) và dấu cách. Xâu không bắt đầu và không kết thúc bằng dấu cách.

Kết quả: Ghi ra tập tin văn bản **LYRICS.OUT** một số nguyên – giá trị k tìm được.

Ví dụ:

| LYRICS . INP | LYRICS . OUT |
|--|--------------|
| 2 1 Vietnam Vietnam go go go Vietnam Vietnam lets go | 3 |

THÁP XU

Alice và Bob chơi trò bốc xu từ tháp được xây dựng bởi n đồng xu. Hai bạn chọn hai số nguyên dương khác nhau K và L . Hai người lần lượt đi. Alice đi trước. Mỗi người, khi đến lượt mình, được bốc khỏi tháp 1, K hoặc L xu. Ai bốc được đồng xu (hoặc các đồng xu) cuối cùng là thắng. Sau rất nhiều lần chơi, Alice nhận thấy rằng có những trường hợp mình chắc chắn thắng không phụ thuộc vào cách đi của Bob, ngược lại, có trường hợp dù đi thế nào Bob vẫn thắng. Trước ván chơi mới Alice nóng lòng muốn biết mình có thắng được hay không.

Yêu cầu: Cho n , K và L . Hãy xác định Alice hay Bob sẽ thắng.

Dữ liệu: Vào từ tập tin văn bản **COINTOWER.INP**

- Dòng đầu tiên chứa 3 số nguyên K , L và m , trong đó m – số ván chơi ($1 < K < L < 10; 3 < m < 50$).
- Dòng thứ hai chứa m số nguyên n_1, n_2, \dots, n_m trong đó $n_i (1 \leq n_i \leq 10^6)$ – số xu trong tháp ở ván chơi thứ i .

Kết quả: Đưa ra tập tin văn bản **COINTOWER.OUT** xâu m ký tự từ tập $\{A,B\}$, ký tự thứ i là A nếu Alice thắng được và là B nếu Bob thắng.

Ví dụ:

| COINTOWER . INP | COINTOWER . OUT |
|-------------------------------|-----------------|
| 2 3 5 3 12 113 25717 88888 | ABAAB |

BIẾN ĐỔI DÃY

Xét dãy số nguyên a_1, a_2, \dots, a_n thỏa $a_i = i$ với $n = 2^{60}$. Ta thực hiện phép biến đổi sau cho tất cả phần tử trong dãy: Với mỗi a_i , tính tổng các chữ số của phần tử này cho đến khi a_i chỉ còn 1 chữ số. Chẳng hạn phần tử $a_{2019} = 2019$ được biến đổi thành: $2019 \rightarrow 12 (= 2 + 0 + 1 + 9) \rightarrow 3 (= 1 + 2)$, nghĩa là sau phép biến đổi $a_{2019} = 3$.

Yêu cầu: Cho q truy vấn, mỗi truy vấn có dạng l_i, r_i . Hãy tính tổng các phần tử từ vị trí l_i đến r_i sau phép biến đổi dãy.

Dữ liệu: Vào từ tập tin văn bản **MAGICSEQ.INP**

- Dòng đầu tiên chứa số nguyên $q (1 \leq q \leq 100)$ – số lượng truy vấn.
- Mỗi dòng trong q dòng tiếp theo chứa 2 số nguyên $l_i, r_i (1 \leq l_i \leq r_i \leq 2^{60})$.

Kết quả: Ghi ra tập tin văn bản **MAGICSEQ.OUT** gồm q dòng, mỗi dòng ghi một số nguyên là kết quả của truy vấn tương ứng trong dữ liệu vào.

Ví dụ:

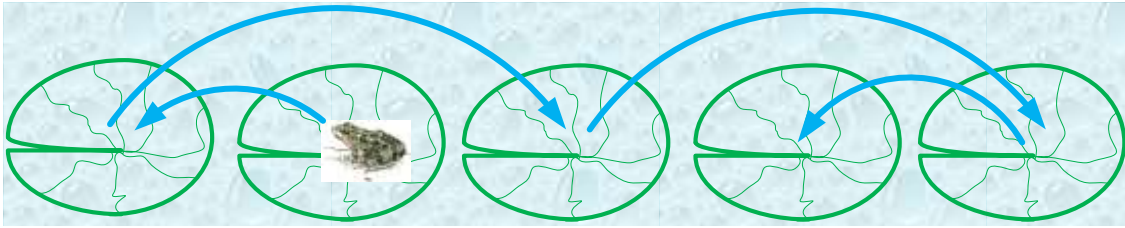
| MAGICSEQ . INP | MAGICSEQ . OUT |
|-----------------------|-----------------------|
| 2 | 38 |
| 8 15 | 9 |
| 2018 2020 | |

Giải thích:

- Test đầu tiên: $a_8 = 8, a_9 = 9, a_{10} = 10, a_{11} = 11, a_{12} = 12, a_{13} = 13, a_{14} = 14, a_{15} = 15$ được biến đổi thành $a_8 = 8, a_9 = 9, a_{10} = 1, a_{11} = 2, a_{12} = 3, a_{13} = 4, a_{14} = 5, a_{15} = 6$.
- Test thứ hai: $a_{2018} = 2018, a_{2019} = 2019, a_{2020} = 2020$ được biến đổi thành $a_{2018} = 2, a_{2019} = 3, a_{2020} = 4$.

CHÚ ẾCH

Chú ếch trong trò chơi đầm lầy phải giải quyết một chuyện phức tạp: có n lá bông súng nổi trên mặt nước, xếp thành một hàng đánh số từ 1 đến n từ trái qua phải ($2 \leq n \leq 1000$). Trên mỗi lá có một con ruồi. Chú ếch đang ngồi ở lá thứ a . Chú có thể nhảy sang lá bên cạnh hay nhảy cách một lá theo hướng bất kỳ. Ếch đã lớn mà lá hãy còn non, vì vậy nó sẽ bị chìm khi chú nhảy khỏi lá. Chú ếch của chúng ta có nhiệm vụ phải bắt hết ruồi và dừng lại ở lá b ($1 \leq a, b \leq n; a \neq b$).



Yêu cầu: Cho n, a và b . Hãy chỉ ra các bước nhảy chú ếch cần thực hiện (-1 là nhảy sang lá bên trái, -2 là nhảy cách một lá sang bên trái; 1, 2 – sang bên phải). Nếu không thể thực hiện được thì đưa ra số 0.

Dữ liệu: Vào từ tập tin văn bản **FROG.INP** chứa 3 số nguyên n, a, b .

Kết quả: Ghi ra tập tin văn bản **FROG.OUT**, mỗi bước nhảy của chú ếch được đưa ra trên một dòng.

Ví dụ:

| FROG . INP | FROG . OUT |
|------------|------------|
| 5 2 4 | -1 |
| | 2 |
| | 2 |
| | -1 |

DÃY SỐ

Xét dãy số nguyên a_1, a_2, \dots, a_n với a_1 được cho trước, các phần tử còn lại được tính theo công thức: $a_i = (a_{i-1})^2 \bmod 10000$.

Yêu cầu: Cho a_1 và n ($0 \leq a_1 \leq 10000$; $1 \leq n \leq 2 \times 10^9$). Hãy xác định a_n .

Dữ liệu: Vào từ tập tin văn bản **SEQ.INP** chứa 2 số nguyên a_1, n .

Kết quả: Ghi ra tập tin văn bản **SEQ.OUT** giá trị a_n .

Ví dụ:

| SEQ . INP | SEQ . OUT |
|-----------|-----------|
| 2 5 | 5536 |

ĐOẠN DƯỜNG

Cho dãy số nguyên $A = (a_1, a_2, \dots, a_n) (|a_i| \leq 10^9)$. Hãy tìm một đoạn dài nhất gồm các phần tử liên tiếp trong dãy $A: (a_L, a_{L+1}, \dots, a_H)$ có tổng là số dương.

Dữ liệu: Vào từ tập tin văn bản **PS.INP**

- Dòng 1: Chứa số nguyên dương $n (n \leq 10^5)$
- Dòng 2: Chứa n số a_1, a_2, \dots, a_n theo đúng thứ tự, đảm bảo có ít nhất một số dương trong dãy.

Kết quả: Ghi ra tập tin văn bản **PS.OUT** hai chỉ số L và H trên một dòng

Ví dụ:

| PS . INP | PS . OUT |
|-----------------------------------|----------|
| 10 -5 -2 -3 4 -6 7 -8 9 -1 -20 | 3 9 |

LUYỆN TẬP THỂ THAO

Một chú thỏ con lên kế hoạch tập luyện thể thao trong n ngày bằng cách chinh phục một cầu thang cao n bậc thang. Ngày đầu tiên, chú thỏ nhảy mỗi bước 1 bậc thang. Sau mỗi ngày sức khỏe chú thỏ tăng lên, mỗi bước nhảy cao hơn 1 bậc thang so với mỗi bước nhảy ở ngày trước đó. Chú thỏ đặt ra mục tiêu mỗi ngày đều phải chinh phục bậc thang thứ n với ít bước nhảy nhất.

Như vậy để hoàn thành mục tiêu, có một số ngày các bước nhảy của chú thỏ đều phải nhảy hết sức, và một số ngày có 1 bước nhảy dưới sức mình. Chẳng hạn với $n = 9$ thì ngày đầu tiên thực hiện 9 bước nhảy hết sức (mỗi bước nhảy 1 bậc thang). Ngày thứ 2 thực hiện 4 bước nhảy hết sức (mỗi bước nhảy 2 bậc thang) và 1 bước nhảy dưới sức (nhảy 1 bậc thang). Ngày thứ 3 thực hiện 3 bước nhảy hết sức (mỗi bước nhảy 3 bậc thang). Nói cách khác, ngày thứ d chú thỏ thực hiện $(n \text{ div } d)$ bước nhảy hết sức, mỗi bước nhảy d bậc thang và 1 bước nhảy dưới sức với $(n \text{ mod } d)$ bậc.

Như vậy với $n = 9$ thì chú thỏ thực hiện tổng cộng $9+4+3+2+1+1+1+1+1=23$ bước nhảy hết sức.

Yêu cầu: Cho số nguyên dương $n (n \leq 10^{12})$. Cho biết toàn bộ quá trình tập luyện n ngày, chú thỏ đã thực hiện tổng cộng bao nhiêu bước nhảy hết sức.

Dữ liệu: Vào từ tập tin văn bản **PRACTICE.INP** số nguyên dương n .

Kết quả: Ghi ra tập tin văn bản **PRACTICE.OUT** một số nguyên là tổng số bước nhảy hết sức của chú thỏ.

Ví dụ:

| PRACTICE . INP | PRACTICE . OUT |
|----------------|----------------|
| 9 | 23 |

BẮN TÊN

Trò chơi “Bắn tên” đang là trò chơi rất được ưa chuộng trên máy tính và các thiết bị di động. Trò chơi được thiết kế khá đơn giản: Có n quả bóng bay đánh số từ 1 tới n xếp theo hàng ngang từ trái qua phải, quả bóng thứ i nằm ở độ cao h_i . Người chơi phải bắn các mũi tên làm vỡ hết các quả bóng. Mỗi mũi tên được bắn theo phương ngang từ trái qua phải ở một độ cao ban đầu tùy chọn. Khi mũi tên chạm vào một quả bóng cùng độ cao nó đang bay, nó sẽ làm vỡ quả bóng đó và tiếp tục bay sang phải với độ cao giảm đi 1 đơn vị.

Yêu cầu: Cho biết số mũi tên ít nhất cần sử dụng để làm vỡ tất cả các quả bóng.

Dữ liệu: Vào từ tập tin văn bản **MINARROWS.INP**

- Dòng 1 chứa số nguyên dương $n (n \leq 10^6)$.
- Dòng 2 chứa n số nguyên dương $h_1, h_2, \dots, h_n (h_i \leq 10^6)$.

Kết quả: Ghi ra tập tin văn bản **MINARROWS.OUT** một số nguyên duy nhất là số mũi tên ít nhất cần sử dụng để làm vỡ tất cả các quả bóng.

Ví dụ:

| MINARROWS . INP | MINARROWS . OUT |
|------------------|-----------------|
| 6 5 6 5 3 2 1 | 3 |

PHÂN TÍCH SỐ

Cho số nguyên dương $n(n \leq 50)$, hãy liệt tất cả cách phân tích n thành tổng các số nguyên dương, các cách phân tích là hoán vị của nhau được tính là 1 cách.

Dữ liệu: Vào từ tập tin văn bản **ANALYSE.INP** chứa số nguyên dương n .

Kết quả: Ghi ra tập tin văn bản **ANALYSE.OUT** các cách phân tích n , mỗi cách phân tích trên một dòng và được ghi theo thứ tự từ điển.

Ví dụ:

| ANALYSE . INP | ANALYSE . OUT |
|---------------|---|
| 3 | 1 1 1 1 2 3 |
| 6 | 1 1 1 1 1 1 1 1 1 1 2 1 1 1 3 1 1 2 2 1 1 4 1 2 3 1 5 2 2 2 2 4 3 3 6 |

XE ĐƯA RƯỚC

Cứ sau k phút lại có một ô tô của một công ty xe buýt qua bến đỗ. Biết rằng thời gian đến bến này của n hành khách. Nếu hành khách đến bến trước hoặc đúng thời điểm ô tô đến thì họ có thể lên xe ngay. Ô tô không bao giờ đợi.

Yêu cầu: Hãy xác định xem ô tô đầu tiên của công ty cần đến bến này vào thời điểm nào để

- Tổng thời gian đợi của tất cả hành khách là nhỏ nhất.
- Thời gian đợi xe lâu nhất của một hành khách là nhỏ nhất.

Dữ liệu: Vào từ tập tin văn bản **BUSES.INP**

- Dòng đầu chứa 2 số nguyên $n, k (k \leq 500; n \leq 10^5)$.
- Dòng tiếp theo là n thời điểm của n khách tới bến.

Kết quả: Ghi ra tập tin văn bản **BUSES.OUT**

- Dòng đầu tiên ghi tổng thời gian đợi của tất cả hành khách và thời điểm đến của xe bus.
- Dòng tiếp theo ghi thời gian đợi xe lâu nhất của một hành khách và thời điểm đến của xe bus.

Ví dụ:

| BUSES . INP | BUSES . OUT |
|-----------------|-------------|
| 5 100 | 91 10 |
| 0 210 99 551 99 | 52 51 |

XEM PHIM

Có n bộ phim sẽ chiếu trên nhiều kênh truyền hình khác nhau. Với mỗi bộ phim, ta biết thời gian bắt đầu chiếu và thời gian kết thúc. Hãy cho biết một người có thể xem được tối đa bao nhiêu bộ phim, mỗi bộ phim phải xem trọn vẹn từ đầu cho đến khi kết thúc.

Dữ liệu: Vào từ tập tin văn bản **FILMS.INP**

- Dòng đầu tiên chứa số nguyên $n(1 \leq n \leq 2 \times 10^5)$ – số lượng bộ phim.
- Mỗi dòng trong n dòng tiếp theo chứa 2 số nguyên $a, b(1 \leq a < b \leq 10^9)$ – thời gian bắt đầu chiếu và thời gian kết thúc của bộ phim tương ứng.

Kết quả: Ghi ra tập tin văn bản **FILMS.OUT** một số nguyên là số lượng phim nhiều nhất xem được.

Ví dụ:

| FILMS . INP | FILMS . OUT |
|------------------------|-------------|
| 3 3 5 4 9 5 8 | 2 |

ĐOẠN CÂN BẰNG

Cho dãy số nguyên dương $A = (a_1, a_2, \dots, a_n)$, một dãy con khác rỗng gồm các phần tử liên tiếp trong A được gọi là đoạn cân bằng của A nếu số số chẵn trong dãy con đúng bằng số số lẻ trong dãy con đó.

Ví dụ với $A = (1, 2, 3, 4, 6)$, có 4 đoạn cân bằng của A là: $(1, 2)$, $(1, 2, 3, 4)$, $(2, 3)$, $(3, 4)$.

Yêu cầu: Đếm số lượng đoạn cân bằng của dãy A .

Dữ liệu: Vào từ tập tin văn bản **PARITY.INP**

- Dòng đầu tiên chứa số nguyên dương n ($n \leq 10^5$).
- Dòng thứ hai chứa dãy gồm n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Kết quả: Ghi ra tập tin văn bản **PARITY.OUT** một số nguyên là số đoạn cân bằng của dãy.

Ví dụ:

| PARITY . INP | PARITY . OUT |
|----------------|--------------|
| 5 1 2 3 4 6 | 4 |

HỆ THỐNG GIÁM SÁT

Hệ thống hỗ trợ kiểm soát sự lây lan của dịch bệnh bằng cách giám sát sự ra vào của người dân tại một địa phương. Hệ thống mã hóa thông tin mỗi người dân bằng một ID là một số nguyên lên đến 18 chữ số.

Hệ thống ghi nhận việc viếng thăm của người dân đến vùng cần theo dõi theo thứ tự thời gian cách đều nhau, các thời điểm được đánh thứ tự từ 1. Dữ liệu này được chuyển đến nhà quản lý nhằm thực hiện các báo cáo cần thiết. Chẳng hạn, nhà quản lý cần biết thời điểm đầu tiên và thời điểm cuối cùng của từng người dân viếng thăm địa phương. Ví dụ hệ thống ghi nhận dữ liệu người dân viếng thăm trong 10 thời điểm như sau: 7 2 10 7 2 6 3 3 3 6.

| Thời điểm | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----------|---|---|----|---|---|---|---|---|---|----|
| ID | 7 | 2 | 10 | 7 | 2 | 6 | 3 | 3 | 3 | 6 |

- Người có ID = 2 có thời điểm thăm lần đầu và lần cuối là 2 và 5
- Người có ID = 3 có thời điểm thăm lần đầu và lần cuối là 7 và 9
- Người có ID = 7 có thời điểm thăm lần đầu và lần cuối là 1 và 4
- Người có ID = 10 chỉ thăm 1 lần tại thời điểm 3.

Yêu cầu: Cho dữ liệu ghi nhận các lần viếng thăm của người dân theo thứ tự thời gian. Hãy cho biết ID của người dân có khoảng cách từ thời điểm viếng thăm đầu tiên đến thời điểm viếng thăm cuối cùng là cách xa nhau nhất.

Dữ liệu: Vào từ tập tin văn bản **MONITOR.INP**

- Dòng đầu tiên chứa số nguyên $n (1 \leq n \leq 10^5)$ – số thời điểm theo dõi.
- Dòng tiếp theo chứa dãy gồm n số nguyên $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^{18})$ – ID của người dân viếng thăm theo thứ tự thời gian.

Kết quả: Ghi ra tập tin văn bản **MONITOR.OUT**

- Dòng đầu tiên ghi ID của người cần báo cáo.
- Dòng tiếp theo ghi một số nguyên là số lượng thời điểm tính từ lần thăm đầu tiên đến lần thăm cuối cùng.
- Nếu có nhiều kết quả thì in ra ID của người đầu tiên trong danh sách.

Ví dụ:

| MONITOR . INP | MONITOR . OUT |
|----------------------------|---------------|
| 10 7 2 19 7 2 6 3 3 3 6 | 6 5 |

ƯỚC CHUNG LỚN NHẤT

Bờm không thích những dãy số mà ước chung lớn nhất của chúng không phải là con số mà Bờm yêu thích. Do đó nếu dãy số này không đúng ý của mình thì Bờm cố gắng điều chỉnh các giá trị trong dãy sao cho ước chung lớn nhất của chúng bằng một số nguyên yêu thích k nào đó của Bờm.

Ở mỗi lần điều chỉnh, Bờm chỉ có thể tăng hoặc giảm giá trị của một phần tử bất kỳ đi 1 đơn vị. Giá trị của phần tử của dãy trước và sau khi điều chỉnh phải đảm bảo là số nguyên dương.

Yêu cầu: Cho dãy số nguyên dương a_1, a_2, \dots, a_n và số nguyên dương k . Tìm số phép điều chỉnh tối thiểu để ước chung lớn nhất của dãy là k hoặc là bội số của k .

Dữ liệu: Vào từ tập tin văn bản **PGCD.INP**

- Dòng đầu tiên chứa số nguyên t ($1 \leq t \leq 10$) – số bộ test.
- Dòng đầu tiên của mỗi bộ test chứa 2 số nguyên dương k, n ($1 \leq k, n \leq 10^6$).
- Dòng thứ hai của mỗi bộ test chứa dãy số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$).

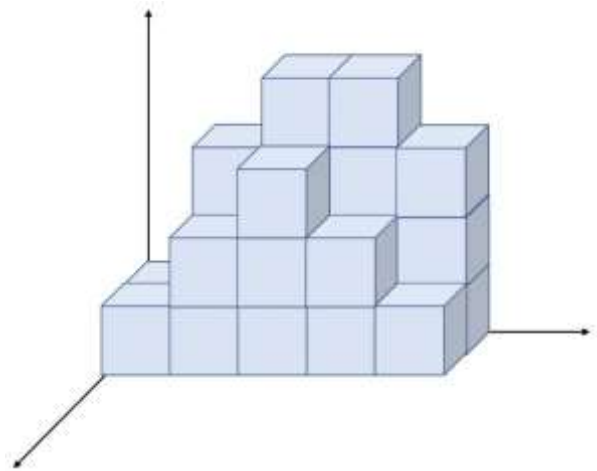
Kết quả: Ghi ra tập tin văn bản **PGCD.OUT** gồm t dòng, mỗi dòng ghi số phép biến đổi tối thiểu của bộ test tương ứng.

Ví dụ:

| PGCD . INP | PGCD . OUT |
|-------------------|------------|
| 1 5 3 4 5 6 | 2 |

DIỆN TÍCH BỀ MẶT

Trong lớp kỹ sư xây dựng nhí, các học sinh được cô giáo phát cho một tấm bìa hình chữ nhật kích thước $H \times W$ dùng để làm mặt bằng cho bài tập thiết kế công trình mà cô giáo sắp giao. Tấm bìa được chia thành $H \times W$ ô vuông đơn vị. Trên mỗi ô vuông của tấm bìa đều được ghi một số nguyên, cụ thể ô ở dòng i cột j ghi số nguyên a_{ij} .



Cô giáo yêu cầu học sinh sử dụng các khối lập phương đơn vị (kích thước $1 \times 1 \times 1$) đặt vừa khít vào các ô vuông trên tấm bìa. Số lượng khối lập phương đặt tại mỗi ô bằng đúng giá trị nguyên được ghi tại ô đó. Các khối lập phương được ráp lại theo hướng dẫn trên tấm bìa tạo thành một mô hình xây dựng.

Yêu cầu: Hãy cho biết tổng diện tích bề mặt của mô hình xây dựng.

Dữ liệu: Vào từ tập tin văn bản **SURFACE.INP**

- Dòng đầu tiên chứa 2 số nguyên H, W ($1 \leq H, W \leq 1000$) – kích thước tấm bìa.
- Dòng thứ i trong H dòng tiếp theo chứa dãy gồm W số nguyên $a_{i1}, a_{i2}, \dots, a_{iW}$ ($0 \leq a_{ij} \leq 10^9$) – dãy số ghi ở dòng thứ i của tấm bìa.

Kết quả: Ghi ra tập tin văn bản **SURFACE.OUT** tổng diện tích bề mặt của mô hình.

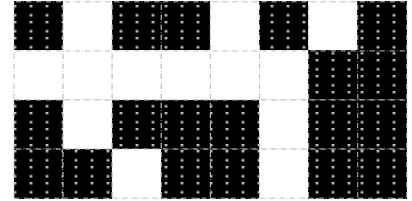
Ví dụ:

| SURFACE . INP | SURFACE . OUT |
|-------------------------------|---------------|
| 2 5 1 3 4 4 3 1 2 3 2 1 | 64 |

CẮT GIẤY

Từ một tấm giấy caro chữ nhật gồm m dòng và n cột, người ta đục một số lỗ, mỗi lỗ đục mất trọn một ô vuông trên tấm giấy. Điều này sẽ làm cho một số ô trên tấm giấy bị tách rời nhau. Hai ô không bị tách rời nhau nếu chúng vẫn còn cạnh chung.

Yêu cầu: Cho kích thước tấm giấy và các ô bị đục lỗ. Hãy tính số lượng mảnh giấy bị tách rời nhau.



Dữ liệu: Vào từ tập tin văn bản **REMSQR.INP**

- Dòng đầu tiên chứa hai số nguyên $m, n (1 \leq m, n \leq 100)$
- Mỗi dòng trong m dòng tiếp theo chứa dãy gồm n kí tự: “#” - tương ứng với ô không bị đục lỗ và “.” – tương ứng với ô lưới bị đục lỗ

Kết quả: Ghi ra tập tin văn bản **REMSQR.OUT** số lượng mảnh bị cắt rời

Ví dụ:

| REMSQR . INP | REMSQR . OUT |
|---|--------------|
| 4 8 # . # # . # . # # # # . # # # . # # # # . # # . # # | 6 |

SỐ JUMPING

Số nguyên không âm x được gọi là Jumping nếu các chữ số liên tiếp của x chênh lệch nhau đúng 1 đơn vị. Tất cả số nguyên có 1 chữ số cũng được gọi là số Jumping. Chẳng hạn các số sau đây được gọi là Jumping: 0, 1, 2, 9, 10, 12, 54, 98, 89, 67656789.

Yêu cầu: Cho số nguyên dương n . Tìm tất cả số Jumping không vượt quá n .

Dữ liệu: Vào từ tập tin văn bản **JNUM.INP**

- Dòng đầu tiên chứa số nguyên $t (1 \leq t \leq 100)$ – số bộ test.
- Mỗi dòng trong t dòng tiếp theo chứa số nguyên $n (1 \leq n \leq 10^9)$.

Kết quả: Ghi ra tập tin văn bản **JNUM.OUT** gồm t dòng, mỗi dòng ghi dãy các số Jumping không vượt quá số nguyên n tương ứng trong dữ liệu vào.

Ví dụ:

| JNUM. INP | JNUM. OUT |
|-----------|---|
| 2 | 0 1 2 3 4 5 6 7 8 9 10 |
| 10 | 0 1 2 3 4 5 6 7 8 9 10 12 21 23 32 34 43 45 |
| 50 | |

ĐƯỜNG ĐI NGẮN NHẤT

Sau một thời gian dài xung đột, 2 bộ tộc Yama và Chacha đã quyết định xóa bỏ mọi mâu thuẫn, quay trở lại sống hòa thuận như xưa. Để bắt đầu cho một giai đoạn mới, người dân 2 bộ tộc cùng nhau xây một con đường nối 2 vùng lãnh thổ với nhau. Bài toán đặt ra là con đường nào ngắn nhất?

Có thể coi lãnh thổ của 2 bộ tộc nằm trong một bảng các ô vuông. Mỗi bộ tộc sở hữu một tập các ô vuông liên thông (có thể đi đến nhau qua các ô vuông chung cạnh trong tập). Ban đầu không có 2 ô vuông nào thuộc 2 bộ tộc có cạnh chung. Con đường cần xây sẽ phải bao gồm một số ô vuông liên thông với nhau và có một ô vuông kề cạnh với một ô vuông thuộc bộ tộc Yama, một ô kề cạnh với một ô vuông thuộc bộ tộc Chacha. Hãy tìm con đường chiếm ít ô vuông nhất.

Dữ liệu: Vào từ tập tin văn bản **PATH.INP**

- Dòng đầu tiên chứa 2 số nguyên $n, m (1 \leq n, m \leq 50)$ là số dòng và cột của bản đồ.
- n dòng tiếp theo, mỗi dòng ghi m kí tự mô tả một dòng của bản đồ, kí tự '.' thể hiện ô trống, kí tự 'X' thể hiện ô thuộc sở hữu của một bộ tộc. Các kí tự 'X' đảm bảo sẽ hình thành 2 vùng liên thông.

Kết quả: Ghi ra tập tin văn bản **PATH.OUT** số lượng ô vuông ít nhất mà con đường cần chiếm.

Ví dụ:

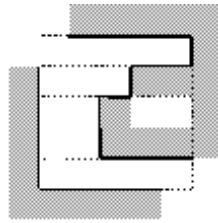
| PATH . INP | PATH . OUT | Giải thích |
|---|------------|---|
| 6 16XXXX. . . .XXX.. ...XXXX. . . .XX.. .XXXX.XXX..XXXXX..XXX. . . | 3 |XXXX. . . .XXX.. ...XXXXo. . . .XX.. .XXXX. .oo. .XXX..XXXXX..XXX. . . |

NHÀ GƯƠNG CƯỜI

Ban quản lý nhà gương cười muốn thay đổi toàn bộ tấm gương phủ tường để phục vụ tốt hơn khách tham quan. Bạn cần giúp ban quản lý tính diện tích gương cần mua để thực hiện công việc đó.

Nhà gương được mô tả bởi bảng ký tự kích thước $n \times n$ ($3 \leq n \leq 33$). Một số ô của bảng chứa dấu chấm ('.') để ký hiệu ô trống. Một số ô khác chứa dấu thăng ('#') để ký hiệu ô vuông được bao bọc bởi các bức tường. Tất cả các ô vuông đều có kích thước 3×3 mét.

Người ta xây dựng các bức tường bao bọc chung quanh nhà gương, ngoại trừ ô ở góc trên trái và ô ở góc dưới phải tương ứng với lối vào và ra của nhà gương. Các bức tường cũng được xây dựng chung quanh các ô đánh dấu thăng. Ngoài ra không có bức tường nào khác. Giả thiết rằng ô ở góc trên trái và dưới phải của bảng luôn chứa dấu chấm.



Nhà gương mô tả trong file dữ liệu

Yêu cầu: Hãy tính diện tích các bức tường ở phía trong của nhà gương (phần nhìn thấy được bởi du khách vào tham quan nhà gương). Chiều cao của mỗi bức tường đều là 3 mét.

Dữ liệu: Vào từ tập tin văn bản **MIRROR.INP**

- Dòng đầu tiên chứa số n .
- Dòng thứ i trong số n dòng tiếp theo chứa n ký tự mô tả các ô ở dòng thứ i của bảng ký tự mô tả nhà gương. Các ký tự trên một dòng chỉ là dấu chấm hoặc dấu thăng được ghi liền nhau.

Kết quả: Ghi ra tập tin văn bản **MIRROR.OUT** diện tích của các bức tường mà bạn tìm được.

Ví dụ:

| MIRROR . INP | MIRROR . OUT |
|---|--------------|
| 5## ..#.. .### | 198 |

NÔNG TRẠI

Trong trại chăn nuôi của John có nuôi một số con gà. Trong khi John đang ngủ say, những con cáo đói đã vào trại và tấn công đàn gà.

Trại có dạng hình chữ nhật gồm các ô được đánh số bởi m hàng và n cột. Mỗi ô chứa một kí tự: kí tự '.' là ô trống, kí tự '#' là hàng rào, kí tự 'c' là gà, kí tự 'f' là cáo. Ta coi 2 ô là cùng một chuồng nếu có thể di chuyển từ ô này sang ô kia bằng đường đi chỉ gồm các đường theo hàng ngang hoặc thẳng đứng mà không bị vướng vào hàng rào.

May thay, những con gà cũng biết tự vệ. Chúng có thể mổ chết những con cáo trong chuồng nếu số lượng gà lớn hơn số lượng cáo trong cùng chuồng. Ngược lại, những con cáo sẽ ăn hết gà trong chuồng đó.

Yêu cầu: Ban đầu, các con gà và các con cáo đã được xác định trong các miền của trại. Hãy tính số lượng gà và số lượng cáo còn lại vào sáng hôm sau.

Dữ liệu: Vào từ tập tin văn bản **FARM.INP**

- Dòng đầu chứa 2 số nguyên dương m, n là số hàng và số cột của trại ($m, n \leq 1000$).
- m dòng tiếp theo, dòng i chứa n kí tự, ký tự thứ j là ký hiệu của ô (i, j) trong trại.

Kết quả: Ghi ra tập tin văn bản **FARM.OUT** một dòng gồm hai số nguyên cách nhau bởi dấu cách: số cáo và số gà còn lại trong trại.

Ví dụ:

| FARM . INP | FARM . OUT |
|--|------------------|
| <pre> 8 8 .##### #..c...# #.####.# #.#f.#.# #.#.c#c# #c.##..# #.#..f.# .##### </pre> | <pre> 1 3 </pre> |

THOÁT MÊ CUNG

Bản đồ một mê cung là một hình chữ nhật kích thước $m \times n$ được chia thành lưới ô vuông đơn vị trong đó ô nằm trên giao điểm của hàng i và cột j được gọi là ô (i, j) . Trên mỗi ô ghi một trong ba ký tự:

- ' . ' (dấu chấm): Nếu ô đó an toàn
- ' # ' : Nếu ô đó không được đi vào
- ' * ' : Nếu là ô có một nhà thám hiểm đang đứng.

Duy nhất chỉ có 1 ô ghi dấu ' * '. Nhà thám hiểm có thể từ một ô đi sang một trong số các ô chung cạnh với ô đang đứng. Một cách đi thoát khỏi mê cung là một hành trình đi ra một ô biên mà không đi vào ô mang dấu ' # '. Hãy chỉ giúp cho nhà thám hiểm một hành trình thoát ra khỏi mê cung. Biết rằng luôn có cách thoát khỏi mê cung đã cho.

Dữ liệu: Vào từ tập tin văn bản **LABYRINTH.INP**

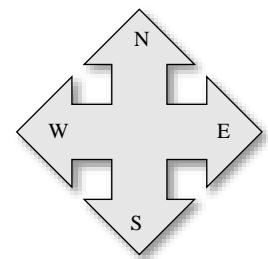
- Dòng đầu chứa hai số nguyên dương $m, n \leq 1000$.
- m dòng tiếp theo, dòng thứ i chứa n ký tự, ký tự thứ j là ký tự ghi trên ô (i, j) của bản đồ.

Kết quả: Ghi ra tập tin văn bản **LABYRINTH.OUT** một dãy ký tự liên tiếp. Ký tự thứ k cho biết hướng đi tại bước di chuyển thứ k . Cụ thể là:

- Ký tự ' N ' : Di chuyển từ một ô sang ô kề cạnh phía trên
- Ký tự ' S ' : Di chuyển từ một ô sang ô kề cạnh phía dưới
- Ký tự ' W ' : Di chuyển từ một ô sang ô kề cạnh bên trái
- Ký tự ' E ' : Di chuyển từ một ô sang ô kề cạnh bên phải

Ví dụ:

| LABYRINTH . INP | LABYRINTH . OUT |
|---|---------------------------------|
| <pre> 8 8 # .##### # .#*...# # .##### # .#....# # .# .#### # .# .#### ## ##### </pre> | <pre> EEESSWWWSSSWNNNNNN </pre> |



HỘI CHỢ

Một khu hội chợ có $m \times n$ gian hàng được bố trí trong một khu hình chữ nhật kích thước $m \times n$. Các hàng của hình chữ nhật được đánh số từ trên xuống dưới bắt đầu từ 1 đến m , còn các cột – đánh số từ trái sang phải, bắt đầu từ 1 đến n , ô nằm giao của hàng i và cột j là gian hàng (i, j) . Mỗi gian hàng trưng bày một sản phẩm và đều có cửa thông với các gian hàng chung cạnh với nó. Khách tham quan đi vào khu hội chợ từ một gian hàng bất kỳ bên trái (i bất kỳ, $j = 1$) và không nhất thiết phải thăm quan tất cả các gian hàng. Khách chỉ có thể đi ra khỏi khu hội chợ từ các gian hàng bên phải (i bất kỳ, $j = n$), tại mỗi gian hàng khách có thể di chuyển qua các gian hàng có cửa thông với nó. Khi đi vào gian hàng (i, j) thì khách tham quan phải mua vé giá là a_{ij} .

Yêu cầu: Tính chi phí ít nhất mà khách tham quan phải trả khi tham quan khu hội chợ.

Dữ liệu: Vào từ tập tin văn bản **FUNFAIR.INP**

- Dòng đầu tiên ghi số $m, n (2 < m, n \leq 200)$
- m dòng sau, mỗi dòng chứa n số nguyên không âm, cho biết giá vé các gian hàng của khu hội chợ. Giá vé tại gian hàng (i, j) là $a_{ij} (0 \leq a_{ij} \leq 30000)$

Kết quả: Ghi ra tập tin văn bản **FUNFAIR.OUT** gồm một số duy nhất là chi phí ít nhất tìm được.

Ví dụ:

| FUNFAIR. INP | FUNFAIR. OUT |
|--------------------------------------|---------------------|
| 3 4 2 1 9 1 5 0 3 4 2 1 9 1 | 10 |

FLAPPY BIRD PHIÊN BẢN DỄ

Steve phát triển một game tương tự Flappy Bird để chơi trên smart phone. Màn hình của game có thể xem như một lưới các ô vuông kích thước $10 \times n$. Chú chim xuất phát từ góc trái dưới và di chuyển sang phải với tốc độ 1 ô vuông/giây.

Người chơi điều khiển chim tránh các chướng ngại vật trên đường đi bằng cách như sau:

- Khi chạm vào màn hình, chim di chuyển theo hướng chéo lên một góc 45^0 với tốc độ 1 ô/giây. Khi đụng trần (dòng 1), nếu vẫn giữ chạm màn hình thì chim tiếp tục bay ngang sang phải.
- Khi thả chạm màn hình, chim di chuyển theo hướng chéo xuống một góc 45^0 với tốc độ 1 ô/giây. Khi chạm sàn (dòng 10), chim tiếp tục di chuyển ngang sang phải.

Khi chim chạm đến cột n thì thắng trò chơi, nhưng điều này không phải lúc nào cũng dễ dàng.

Yêu cầu: Cho sơ đồ mô tả toàn cảnh màn hình trò chơi. Hãy chỉ ra một cách điều khiển để thắng.

Dữ liệu: Vào từ tập tin văn bản **BIRD.INP**

- Dòng đầu tiên chứa số nguyên dương $n (1 \leq n \leq 10^5)$ – kích thước màn hình trò chơi.
- Mỗi dòng trong 10 dòng tiếp theo chứa chuỗi gồm n kí tự '.' hoặc 'X'. Trong đó '.' tương ứng với ô vuông có thể di chuyển được, 'X' tương ứng với ô vuông có chướng ngại vật,.
- Dữ liệu được cho luôn đảm bảo có cách đi để thắng.

Kết quả: Ghi ra tập tin văn bản **BIRD.OUT**

- Dòng đầu tiên là số nguyên $p (0 \leq p \leq 50000)$ – số thao tác điều khiển chạm màn hình.
- Mỗi dòng trong p dòng tiếp theo ghi 2 số nguyên t, p . Trong đó t là thời điểm chạm (tính từ lúc bắt đầu) và d là thời gian giữ chạm. Các thao tác đưa theo trình tự thời gian.

Ví dụ:

| BIRD . INP | BIRD . OUT | Giải thích |
|---|-----------------|---|
| 11XX...XXX...XX ...XX...XX.XXX....X..... ...XX...X. ...XX...XX. ...X...XX.. | 2 1 4 7 2 |XX...XXX...XX ...XX...XX.XXX....*...*.*X*.*.* ...*XX.*.X. ..*XX...XX. ** .X...XX.. |

TRUY TÌM KHO BÁU

Gỡ mìn là một trò chơi logic mà các học sinh thường hay tranh thủ chơi trong các giờ tin học tại phòng máy. Đoàn trường tổ chức một trò chơi mô phỏng game dò mìn tại sân trường nhưng được cải tiến không những dò mìn mà còn dò cả kho báu nữa.

Ban tổ chức bố trí sân trường thành một hình chữ nhật gồm các ô vuông đơn vị. Mỗi ô vuông sẽ được giấu một báu vật tương ứng với một điểm thưởng p nguyên dương hoặc một quả mìn (dĩ nhiên là mìn giả rồi).

Người chơi sẽ di chuyển để thu gom báu vật và phải tránh những ô có mìn, nếu không sẽ bị mất lượt. Người chơi có thể di chuyển qua lại giữa các ô chung cạnh. Bắt đầu mỗi lượt chơi, người chơi được phép xuất phát tại một ô bất kì trên sân trường, nếu đó là ô chứa mìn thì bị mất lượt chơi, ngược lại sẽ nhận được một điểm thưởng tương ứng với báu vật đặt tại ô đó.

Bòm vô tình biết được bản đồ mô tả vị trí đặt báu vật và bốc thăm trúng thưởng được k lượt chơi. Bòm tự hỏi rằng với k lượt chơi thì cậu ta có thể đạt tổng số điểm thưởng cao nhất là bao nhiêu.

Dữ liệu: Vào từ tập tin văn bản **TREASURE.INP**

- Dòng đầu tiên chứa 2 số nguyên dương n, k ($1 \leq n, k \leq 10^6$) – số ô có báu vật và số lượt chơi.
- Dòng thứ i trong n dòng tiếp theo chứa 3 số nguyên dương r_i, c_i, p_i ($1 \leq r_i, c_i \leq 1000; 1 \leq p_i \leq 10^6$).
Trong đó r_i, c_i là tọa độ nơi chứa báu vật, p_i là điểm thưởng tương ứng.

Kết quả: Ghi ra tập tin văn bản **TREASURE.OUT** số điểm thưởng cao nhất có thể đạt được.

Ví dụ:

| TREASURE . INP | TREASURE . OUT |
|--|----------------|
| 10 3 7 10 19 1 1 5 101 1 1 2 2 8 102 1 5 7 11 8 200 202 30 2 1 2 3 2 4 103 1 7 | 76 |