TỔNG HỢP BÀI DAY 1

Bài 1: Rắn săn mồi - SNAKEANDAPPLE (0.5 s, 1G)

An rất nghiện chơi rắn săn mồi. Một hôm, An chợt nảy ra một bài toán: Cho một lưới M x N ô (hàng được đánh số từ 1 đến M, cột được đánh số từ 1 đến N), là nơi chú rắn sẽ di chuyển để tìm những quả táo ngon lành, hãy tìm cách tối ưu nhất (đường đi ngắn nhất, nếu độ dài bằng nhau, chọn đường có số lần rẽ ít nhất) để chú rắn có thể ăn hết những quả táo trên lưới ô vuông đó? Biết chú rắn không dài ra khi ăn táo.

Input: Dòng đầu: Hai số M, N

M dòng sau, mỗi dòng N kí tự thể hiện:

. là ô trống

A là ô chứa táo

W là tường (Rắn không thể đi xuyên qua tường)

S là vị trí chú rắn (Chỉ có duy nhất 1 kí tự S trong bảng)

Output: Dòng đầu: in ra độ dài đường đi và số lần rẽ.

Dòng sau: in ra cách đi đó (Biết ^ là đi lên, < là đi trái, > là đi phải, v là đi xuống)

Nếu không thể đi theo quy tắc hoặc không thể ăn hết táo, in ra -1.

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 4 7  . . A W A W S  A . . . . A .  . A W . . . .  . . . . W . A | 17 9  v v v < ^ ^ < ^ v < < < < ^ > v v |
| 3 5  . . W A W  S . A W .  . . A A W | -1 |

Subtasks:

Subtask 1 (10 điểm): M, N <1000, A<15, không có ô tường

Subtask 2 (90 điểm): 1000 < M, N < , 15<A<

“An và mối tình đầu của cậu ta, rẽ trái rẽ phải y như con rắn săn mồi mà vẫn không tới đích, kết quả -1.”

Bài 2: Con số may mắn của An - LUCKYNUMBER (1.0 s, 4G)

An sau khi bị người yêu đá, quyết định đến gặp một “*bói thầy”*

*“Bói thầy*” chỉ đưa cho An T bộ 3 số gồm ba số nguyên N, L, R và bảo:

Để tìm được giá trị may mắn của bạn, bạn cần phải tính tổng những số có tổng bình phương tất cả các chữ số là số nguyên tố và chữ số lớn nhất của số đó cũng phải là một số nguyên tố. Và số đó không bé hơn L và không lớn hơn R. Hãy gọi tổng đó là S. Giờ hãy gộp tất cả các số như vậy thành một dãy A, hãy tìm phần tử thứ N của dãy đó. Gọi số đó là K. Bây giờ hãy lấy S chia dư K. Đó gọi là “giá trị may mắn của ba số”. Hãy thực hiện như thế với tất cả các bộ ba số N, L, R mà ta đưa cho. Bây giờ hãy tính trung bình cộng tất cả các “giá trị may mắn của ba số” (làm tròn đến chữ số thứ 18 ở hàng thập phân). Đó chính là con số may mắn của cậu đó An à!

Vì có quá nhiều bộ ba số nên An không thể tính hết “đường tình duyên” của mình. Bạn hãy giúp An xem còn mối tình nào đang chờ An ở tương lai phía trước nhé!

Input: Dòng đầu: Số T – thể hiện số lượng bộ ba số.

T dòng sau, mỗi dòng ba số N, L, R

Output: In ra một dòng duy nhất chính là con số may mắn của An.

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  1 1 1 | 1 |

Subtasks:

Subtask 1 (10 điểm): N < 1000, L, R < 500, T < 50

Subtask 2 (90 điểm): 1000 < N < , 500 < L, R < , 50 < T <

“Để tôi tiết lộ cho bạn một bí mật, nhân vật *bói thầy* là một nhân vật **chủ chốt** ở Day 2, và cũng được gọi với một cái tên bắt đầu bằng chữ A.”

Bài 3: Tetris - CLEARTETRIS (3.0s, 3G)

“An đã tìm được lối ra trong mê cung rắn săn mồi, nhưng An giờ lại muốn làm fuho, xếp gạch, không mún iu em…”

Sau khi bị người yêu chia tay vì quá nghiện rắn săn mồi, An đã bỏ thành công game đó và có người yêu mới. Nhưng An lại bị nghiện Tetris làm cho người yêu mới bỏ An mới đau chứ! Hôm nay, An “lại” chợt nảy ra một bài toán:

Cho T lưới M x N và K quân tetris. Hãy tính số hàng bạn có thể hoàn thành nhiều nhất dựa trên quy tắc của trò chơi Tetris truyền thống (kí hiệu xoay: > là xoay phải 90 độ một lần, < là xoay trái 90 độ một lần)

Input: Dòng đầu: số T – thể hiện số testcase

Mỗi testcase có cách thức như sau:

Dòng đầu: ba số M, N, K

Những dòng sau: hình dáng các quân tetris (được ngăn cách bởi hai dấu cách, các quân tetris được thả theo thứ tự từ trái qua phải)

M dòng cuối, mỗi dòng N kí tự như sau:

. là ô trống

x là ô đã có những quân tetris khác thế chỗ

Output:

Với mỗi testcase, in ra cột trái cùng của vị trí thả của những quân tetris trên cùng một dòng sao cho số dòng hoàn thành được là nhiều nhất ở dòng đầu tiên và số hàng hoàn thành được ở dòng thứ hai (1-indexed).

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  7 6 4  x x x x x x x x x  x x x x x  . . . . . .  . . . . . .  . . . . . .  . . . . . .  . . . . . .  x x . . . x  x x . x x x | 3 3 5 1  3 |
| 1  4 4 1  x x  x  x  . . . .  . . . .  x x . x  x x . x | 2  2 |

Subtasks:

Subtask 1 (10 điểm) : K < 50, M, N < 100, T < 5

Subtask 2 (25 điểm) : 49 < K < , 99 < M, N < , 4 < T <

Subtask 3 (50 điểm) : < K < , < M, N < , 999 < T <

Subtask 4 (15 điểm) : < K < , < M, N < , < T <

“An tuy có thể clear đến hàng tetris, nhưng chẳng thể clear 1 dòng ký ức về em. Giờ em đã thuộc về người khác, còn An vẫn ngồi đây, làm fuho trong cái màn hình bé xíu, và chờ em…”

Bài 4: Càng yêu càng âm - NEGSEQ (1.0s, 4G)

Các “nhà khoa học” – cụ thể là “bạn than thiết” của An, đã vẽ ra một dãy số thể hiện tình cảm của An trước, trong và sau khi bị những người yêu đá.

Thật bất ngờ, dãy số tình yêu của An trông như thế này:

-1, -1, -3, -3, -5, -17, -21, -37, -779, -815, -1593, -1298297,…

Hãy trả lời T câu hỏi sau, mỗi câu hỏi có dạng như sau:

Tìm số thứ N của dãy trên và tính “tổng” tất cả các số trong đoạn con (L, R) (0-indexed)

Input: Dòng đầu: số T – thể hiện số testcase

T dòng sau: mỗi dòng 3 số N, L, R

Output: Với mỗi testcase, in ra số thứ N trong dãy số trên và “tổng” các số trong đoạn con (L, R)

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  3 1 2 | -3 -2 |

Subtasks:

Subtask 1 (5% điểm) : T < 5, L, R < 1000, N < 1000000

Subtask 2 (10% điểm) : 5 < T < 1000, 1000 < L, R < , 1000000 < N <

Subtask 3 (30% điểm) : 1000 < T <, < L, R <, < N <

Subtask 4 (55% điểm) : < T < , , L, R < , < N <

“Và “tình yêu” *kết thúc rồi* khi An *không* nhìn vào mắt nàng  
An thấy tim mình, **KHÔNG** tái sinh sau **CỰC** nhiều cơn đau”

Bài 5: Hôm đó em chia tay anh còn anh chia xâu - CUTSTRING (1.0s, 4G)

Sau khi bị người yêu đá 2 lần chỉ vì game, An đã mất hết niềm tin vào tình yêu. An quyết định chia xâu như một lời từ biệt cuối cùng với tình yêu và sự “ngọt ngào” giả dối của nó.

Cho T xâu S với độ dài là L cho trước. Với mỗi xâu S, An sẽ chia nó thành các xâu đối xứng, xâu đối xứng vòng hoặc xâu cân bằng (Xâu đối xứng là một xâu viết từ trái sang phải hay phải sang trái đều như nhau, xâu đối xứng vòng là một xâu mà khi ta dịch các kí tự trong xâu sang phải và kí tự phải cùng dịch về vị trí đầu tiên một số lần nhất định thì sẽ nhận được một xâu đối xứng. Ví dụ xâu ‘array’ là một xâu đối xứng vòng vì khi ta biến đổi xâu ‘array’ => ‘yarra’ => ‘ayarr’ = > ‘rayar’ là một xâu đối xứng. Xâu cân bằng là một xâu có thể chia thành ít nhất 2 đoạn con giống nhau. Ví dụ: xâu ABCABC là xâu cân bằng). Chi phí để chia một đoạn con lần đầu là K, lần tiếp theo là K + C, lần sau nữa là K + 2C, cứ thế cộng dồn. Mục tiêu của An là chia sao cho tất cả các đoạn con đều là đối xứng, đối xứng vòng hoặc cân bằng và với chi phí nhỏ nhất.

Input: Dòng đầu: số T – thể hiện số testcase

Mỗi testcase có cách thức như sau:

Dòng đầu: ba số K, C, L trên cùng một dòng

Dòng sau: xâu S

Output: Với mỗi bộ test, in ra như sau:

Dòng đầu: Số đoạn con chia được, số lần chia và chi phí tối thiểu

Dòng sau: Vị trí của các lần chia (0-indexed)

Ví dụ:

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  4 1 5  AABCB | 2 1 4  1 |

Subtasks:

Subtask 1 (10% điểm) : T < 5, L < 50, K, C < 100

Subtask 2 (25% điểm) : 5 < T < 100, 50 < L < 500, 100 < K, C < 10000

Subtask 3 (50% điểm) : 100 < T < 1000000, 500 < L < 5000000, 10000 < K, C <

Subtask 4 (15% điểm) : 1000000 < T < 20000000, 5000000 < L < , < K, C <

“Khi mọi niềm tin vào tình yêu đã tan biến, An chỉ còn lại chuỗi ký tự dài vô tận... và bài toán tối ưu chia nó thành từng mảnh hồi ức cay đắng. Càng chia nhỏ, chi phí càng lớn – cũng như càng cố quên, càng đau thêm. ”

Bài 6: Ma trận truy vấn – QUERYMATRIX (3.0s, 3G)

“ Nỗi nhớ em trong đêm thật dài  
Thêm lý do cho anh tồn tại  
Để lại chạm vào bờ môi ấy dịu dàng  
Lời thì thầm ngọt ngào bên tai”

An từng hỏi crush rất nhiều điều...  
Nhưng câu trả lời duy nhất mà An nhận lại là... **“…”**

Có một dãy/mạng/lưới nào đó, với rất nhiều truy vấn yêu cầu thay đổi, hỏi han, xóa bỏ, chèn thêm...  
Mỗi truy vấn là một lần đau.

An không biết mình đang sửa code hay sửa lòng.

**“Truy vấn... như những câu hỏi không hồi kết.”**

Cho T ma trận M x N, mỗi ô là một số nguyên, hãy xử lý P truy vấn mỗi ma trận và in ra ma trận sau khi thao tác truy vấn. (0 – indexed)

Các loại truy vấn:

A *x y z*: Cộng z vào ô (x, y)

S *x y z*: Trừ z vào ô (x, y)

AR *x y*: Cộng y vào hàng x

SR *x y*: Trừ y vào hàng x

AC *x y*: Cộng y vào cột x

SC *x y*: Trừ y vào cột x

CR *x y*: Hoán đổi hàng x và y

CC *x y*: Hoán đổi cột x và y

CH *x y z t*: Hoán đổi hai ô (x, y) và (z, t)

DELR *x*: Xóa hàng x

DELC *x*: Xóa cột x

ADDR <M số nguyên> *x*: Thêm hàng tại vị trí x

ADDC <N số nguyên> *x*: Thêm cột tại vị trí x

PR *x*: In hàng x

PC *x*: In cột x

P *x y*: In ô (x, y)

PSR *x*: In tổng hàng x

PSC *x*: In tổng cột x

PMAX: In ra giá trị của ô lớn nhất trong ma trận

PMIN: In ra giá trị của ô bé nhất trong ma trận

PNMAX: In ra tọa độ của ô lớn nhất theo kiểu <*x y*>

PNMIN: In ra tọa độ của ô bé nhất theo kiểu <*x y*>

PMAXR *x*: In ra giá trị của ô lớn nhất trên hàng x

PMINR *x*: In ra giá trị của ô bé nhất trên hàng x

PMAXC *x*: In ra giá trị của ô lớn nhất trên cột x

PMINC *x*: In ra giá trị của ô bé nhất trên cột x

PNMAXR *x*: In ra tọa độ của ô lớn nhất trên hàng x theo kiểu <*x y*>

PNMINR *x*: In ra tọa độ của ô bé nhất trên hàng x theo kiểu <*x y*>

PNMAXC *x*: In ra tọa độ của ô lớn nhất trên cột x theo kiểu <*x y*>

PNMINC *x*: In ra tọa độ của ô bé nhất trên cột x theo kiểu <*x y*>

Input: Dòng đầu: số T – thể hiện số testcase

Mỗi testcase có cú pháp như sau:

Dòng đầu: Ba số M, N, P

M dòng sau, mỗi dòng N số nguyên dương

P dòng cuối, mỗi dòng 1 truy vấn

Output: In ra các ô theo yêu cầu của truy vấn và cuối cùng, in ra ma trận sau tất cả truy vấn.

Ví dụ

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1  4 3 3  1 1 1  2 5 0  -2 3 6  4 2 -4  PR 2  AC 1 3  SC 2 1  PR 2 | 2 5 0  5 4 0  4 0 1  5 4 0  1 2 6  7 1 -4 |

Subtasks:

**Subtask 1** (10 điểm):

* Giới hạn:
  + T <= 10
  + M, N <= 10
  + P <= 100
* Các thao tác cho phép: A *x y z*, S *x y z*, AR *x y*, SR *x, y*, AC *x y*, SC *x y*

**Subtask 2** (15 điểm):

* Giới hạn:
  + T <= 100
  + M, N <= 100
  + P <= 1000
* Các thao tác cho phép thêm: CR *x y*, CC *x y*, CH *x y z t*, DELR *x*, DELC *x*

**Subtask 3** (20 điểm):

* Giới hạn:
  + T <= 500
  + M, N <= 200
  + P <= 10^4

Các thao tác cho phép thêm: ADDR <M số nguyên> *x*, ADDC <N số nguyên> *x*, PR *x*, PC *x*, P *x y*

**Subtask 4** (25 điểm):

* Giới hạn:
  + T <= 10^4
  + M, N <= 10^4
  + P <= 5\*10^5
* Các thao tác cho phép thêm:

PSR *x*, PSC *x*, PMAX, PMIN, PNMAX, PNMIN

**Subtask 5** (30 điểm):

* Giới hạn:
  + T <= 10^6
  + M, N <= 10^9
  + P <= 2\*10^8
* Ma trận ban đầu rỗng, được tạo dần bằng thao tác
* Cần xử lý tối ưu I/O, bộ nhớ và thời gian
* Thêm tất cả truy vấn còn lại