

Mục lục

| | |
|--|---|
| Trạm xăng — FUEL | 2 |
| SSUBSTR | 3 |
| Trò chơi trên đa giác — polygame | 4 |

Bài 1. Trạm xăng — FUEL

Sau vài tháng không thể ra ngoài vì COVID-19, giờ đây Tiến đã có thể đi chơi thỏa thích. Tiến có một chiếc xe máy kỳ diệu, có thể chứa được lượng xăng tùy thích và cứ 1 lít xăng đi được 1 km. Trước khi rời khỏi nhà, Tiến đã đổ vào chiếc xe của mình F lít xăng.

Tiến muốn đi thăm một người bạn ở cách Tiến D km. Trên đường đi có N trạm xăng để Tiến đổ thêm xăng; trạm xăng thứ i cách nhà Tiến X_i km. Ở trạm xăng thứ i , Tiến có thể nạp thêm A_i lít xăng, nhưng chỉ khi lượng xăng hiện tại trên xe không quá B_i .

Yêu cầu: Hãy tìm lượng xăng ban đầu F nhỏ nhất sao cho Tiến có thể có đủ xăng để đến thăm bạn của mình.

Dữ liệu vào

- Dòng thứ nhất chứa hai số nguyên dương N và D ($N \leq 3 \times 10^5, D \leq 10^9$).
- Mỗi dòng trong số N dòng tiếp theo chứa ba số nguyên dương X_i, A_i, B_i ($X_i < D, A_i, B_i \leq 10^9$).

Kết quả

In ra một số nguyên dương duy nhất là đáp án của bài toán.

Ví dụ

| test | answer |
|---|--------|
| 5 100 50 30 25 50 40 25 25 25 25 75 20 25 5 5 25 | 20 |

Hạn chế

- Subtask 1 (50%): $N \leq 10^4$;
- Subtask 2 (50%): Không có ràng buộc gì thêm.

Bài 2. SSUBSTR

Cho N xâu không rỗng gồm các chữ cái latin viết thường s_1, s_2, \dots, s_n .

Gọi S_i là tập các xâu con liên tiếp của s_i . Ví dụ xâu "xyz" có tập các xâu con liên tiếp là $\{ "x", "y", "z", "xy", "yz", "xyz" \}$, xâu "aa" có tập các xâu con liên tiếp là $\{ "a", "aa" \}$.

Yêu cầu: Gọi t_i là số phần tử thuộc S_i nhưng không thuộc bất cứ tập S_j nào mà $j \neq i$. Hãy tính mảng t .

Dữ liệu vào

- Dòng đầu tiên số nguyên dương N là số lượng xâu đề bài cho ($N \leq 10^5$).
- Dòng thứ i trong N dòng tiếp theo là xâu s_i . Tổng độ dài N xâu không vượt quá 10^6 ($\sum |s_i| \leq 10^6$).

Kết quả

- In ra N dòng, dòng thứ i là giá trị của t_i .

Ví dụ

| test | answer |
|--------|--------|
| 3 | 3 |
| abc | 11 |
| debbc | 19 |
| fghhig | |

Hạn chế

- Subtask 1: Tổng độ dài các xâu s_i không vượt quá 2000.
- Subtask 2: $N = 1$.
- Subtask 3: Không có ràng buộc gì thêm.

Bài 3. Trò chơi trên đa giác — polygame

File dữ liệu vào: Gọi hàm
File kết quả: Trả về

Thi đấu với máy tính trong các trò chơi đối kháng luôn tạo ra nhiều cảm hứng cho người chơi. Lần này Hùng gặp phải một trò chơi hóc búa với máy tính sau đây.

Trò chơi diễn ra trên một đa giác lồi n đỉnh được đánh số từ 1 đến n theo chiều kim đồng hồ, trong đó người ta đã vẽ m đường chéo sao cho không có hai đường chéo nào cắt nhau ở trong đa giác và không có 3 đỉnh nào đôi một có đoạn nối (có thể là cạnh hoặc đường chéo của đa giác). Hai đầu thủ luân phiên thực hiện nước đi. Đến lượt mình, người chơi phải vẽ thêm một đường chéo của đa giác sao cho đường chéo này không cắt bất cứ đường chéo nào đã vẽ trước đó tại điểm nằm trong đa giác và đồng thời không có bất cứ tam giác nào được tạo ra (nghĩa là không xuất hiện 3 đỉnh nào của đa giác đôi một có đoạn nối). Người đến lượt mà không có cách thực hiện được lượt đi là người thua cuộc và tất nhiên, đối thủ của anh ta là người giành phần thắng.

Yêu cầu: Hùng được quyền lựa chọn là người thực hiện nước đi trước hay sau, hãy giúp Hùng lựa chọn quyền thực hiện nước đi trước hay sau và thực hiện các lượt chơi để giành phần thắng.

Chương trình phải sử dụng một thư viện riêng. Thư viện bao gồm các file sau: pgamelib.pas (cho Pascal), pgamelib.h (cho C++). Trong chương trình của bạn cần khai báo các thư viện này ở đầu chương trình:

- uses pgamelib; đối với Pascal;
- #include "pgamelib.h" đối với C++.

Thư viện cung cấp các hàm sau:

- Các hàm khởi tạo trò chơi
 - procedure get(var n:longint; var m:longint); đối với Pascal,
 - void get(int &n, int &m); đối với C++.Chương trình phải gọi hàm này để khởi tạo trò chơi. Hàm này trả về một giá trị n là số lượng đỉnh của đa giác và giá trị m là số lượng đường chéo đã nối của đa giác ban đầu. Hàm này chỉ được gọi một lần duy nhất.
- procedure get_uv(var u:longint; var v:longint); đối với Pascal, void get_uv(int &u, int &v); đối với C++.
- Chương trình phải gọi hàm này m lần để lấy về chỉ số hai đầu mút của m đường chéo đã vẽ trong đa giác ban đầu.
- Hàm thực hiện lượt chơi procedure play(i:longint; j:longint; var u:longint; var v:longint); đối với Pascal, void play(int i, int j, int &u, int &v); đối với C++.
- Bạn dùng thủ tục này đưa vào i, j là hai đầu mút của đường chéo được nối trong lượt chơi của mình và thủ tục sẽ trả về u, v là hai đầu mút của đường chéo được nối trong lượt chơi tiếp theo của máy. Nếu bạn lựa chọn chơi trước thì bước đầu tiên gọi hàm play(i, j, u, v) với i, j là nguyên dương. Nếu bạn lựa chọn chơi sau thì bước đầu tiên gọi hàm play($0, 0, u, v$). Chương trình sẽ kết thúc nếu bạn vi phạm luật chơi hoặc không thể thực hiện tiếp lượt chơi.

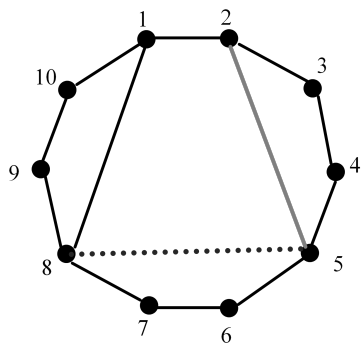
Bạn có thể xem các file được cung cấp trên hệ thống để hiểu rõ hơn về cách tương tác với hệ thống.

Ví dụ

| Gọi hàm | Trả về |
|---|--|
| get(n, m) get_uv(u, v) play($0, 0, u, v$) play($5, 8, u, v$) | $n=10$; $m=1$; $u=5$; $v=10$; $u=4$; $v=9$; $u=0$; $v=0$; Kết thúc chương trình. Bạn đã giành phần thắng. |

Giải thích

Giải thích: Hình dưới đây minh họa cho ví dụ, trong đó các đoạn đen đậm là thông tin đầu vào. Máy đi trước, máy thực hiện việc vẽ cạnh (2,5). Đến lượt mình, người chơi vẽ cạnh (5,8) và giành phần thắng.



Hạn chế

- Subtask 1 (1 điểm): $n \leq 10^6$; n chẵn và không có đường chéo nào ban đầu được nối.
- Subtask 2 (1 điểm): $n \leq 20$.
- Subtask 3 (1.5 điểm): $n \leq 1000$.
- Subtask 4 (1.5 điểm): $n \leq 100000$.