



Ngôn ngữ lập trình C++

§08§ BÀI TẬP LẬP TRÌNH NÂNG CAO



1☀. Bước nhảy (Hard Version)

Bạn cần điều khiển một con robot dịch chuyển trên một trục tọa độ. Ban đầu, robot ở tại vị trí tọa độ 0 (gốc tọa độ). Bàn phím điều khiển có hai nút lệnh: nút “Lùi” và nút “Tiến”. Nếu hiện tại robot đang ở vị trí tọa độ Y , và đã bấm được k lần (ban đầu $k = 0$).

- Khi bấm vào nút “Lùi” nó sẽ dịch chuyển đến tọa độ $Y - 1$.
- Khi bấm vào nút “Tiến” nó sẽ dịch chuyển đến tọa độ $Y + k + 1$.

Chú ý là, sau mỗi lần bấm, số k sẽ tăng lên 1.

Yêu cầu: Hãy tìm cách bấm với số lần ít nhất để đưa robot đến vị trí có tọa độ là X .

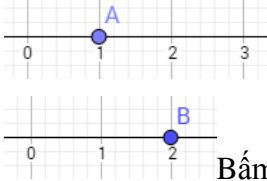
Dữ liệu cho trong file **JumpXH.Inp** gồm:

- Dòng đầu ghi số nguyên dương T ($T \leq 10$) là số bộ dữ liệu.
- T dòng tiếp theo, mỗi dòng ghi một số nguyên dương X ($X \leq 10^6$).

Kết quả ghi ra file **JumpXH.Out** gồm T nhóm dòng, mỗi nhóm dòng gồm:

- Dòng đầu ghi số N là số lần bấm ít nhất để di chuyển robot đến tọa độ X trong bộ dữ liệu tương ứng.
- Dòng thứ 2 ghi chuỗi N kí tự mô tả lần lượt N lần bấm, nếu bấm vào nút “Tiến” thì ghi kí tự ‘T’, nếu bấm vào nút “Lùi” thì ghi kí tự ‘L’. Nếu có nhiều cách bấm thì ghi ra cách bấm bất kì.

Ví dụ:

| JumpXH.Inp | JumpXH.Out | Giải thích |
|------------|------------|---|
| 2 | 1 |  Bấm 1 lần vào nút “Tiến”. |
| 1 | T | |
| 2 | 3 | |
| | TLT | Bấm nút “Tiến”, “Lùi”, “Tiến”. |



2☀. Xếp công việc (Hard Version 1)

Có N công việc được đánh số thứ tự từ 1 đến N . Công việc thứ i có thời gian hoàn thành là A_i và hạn cuối cần hoàn thành là B_i . Nếu công việc thứ i được thực hiện xong trước hoặc ngay tại thời điểm B_i thì ta nói công việc i hoàn thành đúng thời hạn. Ngược lại công việc i không hoàn thành đúng thời hạn.

Yêu cầu: Cho phép thay đổi hạn cuối hoàn thành của 1 công việc (thay B_i). Xét xem, có thể thực hiện đúng thời hạn của tất cả các công việc hay không? Công việc đầu tiên có thể thực hiện từ thời điểm 0. Không được thực hiện nhiều hơn một công việc tại một thời điểm, thực hiện xong công việc



này rồi mới chuyển sang công việc khác. Thời điểm hoàn thành công việc này và thời điểm bắt đầu công việc khác có thể bằng nhau. Công việc đầu tiên có thể bắt đầu từ thời điểm 0.

Dữ liệu cho trong file **WorkH.Inp** gồm:

- Dòng 1 ghi số nguyên dương T ($T \leq 10$) là số bộ dữ liệu:

Mỗi bộ dữ liệu có dạng:

- Dòng thứ nhất ghi số nguyên dương N là số công việc.
- N dòng sau, dòng thứ i ghi hai số nguyên A_i và B_i .

Kết quả ghi ra file **WorkH.Out** gồm T dòng. Mỗi dòng ghi “Yes” nếu bộ dữ liệu tương ứng có thể hoàn thành tất cả các công việc, ngược lại ghi “No”.

Ví dụ:

| WorkH.Inp | WorkH.Out | Giải thích |
|-----------|-----------|---|
| 2 | No | - Bộ dữ liệu 1 không có cách hoàn thành mọi công việc. |
| 4 | Yes | - Bộ dữ liệu 2, thay đổi thời gian hoàn thành của công việc 2 là $B_2 = 20$. Khi đó ta có thể làm việc 1 rồi đến việc 2. |
| 2 2 | | |
| 3 3 | | |
| 3 4 | | |
| 8 10 | | |
| 2 | | |
| 2 2 | | |
| 2 3 | | |

Giới hạn:

- $3 \leq N \leq 10^5$;
- $1 \leq A_i \leq B_i \leq 10^9$.



3. Trò chơi chọn từ

Hai bạn anh em Tôm và Tép hay chơi trò chơi chọn từ như sau: Anh Tôm có N từ, mỗi từ gồm các kí tự latin thường có độ dài không quá 100. Kí tự đầu tiên của mỗi từ hoặc là kí tự ‘a’, hoặc là kí tự ‘b’ hoặc là kí tự ‘c’. Trò chơi được diễn ra K lần, mỗi lần em Tép đưa ra một kí tự thuộc tập 3 kí tự {‘a’, ‘b’, ‘c’}, anh Tôm phải đưa ra một từ trong tập N từ sao cho:

- Kí tự đầu của từ trùng với kí tự mà Tép đưa ra.
- Số lần xuất hiện của từ đó trong tập các từ đã đưa ra là ít nhất.
- Từ đưa ra có số hiệu nhỏ nhất.

Dữ liệu đảm bảo trong N từ luôn có từ bắt đầu là kí tự ‘a’, có từ bắt đầu là kí tự ‘b’, bắt đầu là kí tự ‘c’.

Dữ liệu cho trong file **WordGame.Inp** gồm:

- Dòng đầu ghi hai số nguyên dương N và K ($N, K \leq 100000$).
- N dòng sau, mỗi dòng ghi một từ.
- K dòng cuối cùng, mỗi dòng ghi một kí tự thuộc tập {‘a’, ‘b’, ‘c’}.

Kết quả ghi ra file **WordGame.Out** gồm K dòng, mỗi dòng là một từ mà Tôm cần đưa ra khi em Tép đưa ra một kí tự.



Ví dụ:

| WordGame.Inp | WordGame.Out |
|--------------|--------------|
| 5 5 | aa |
| ab | ba |
| aa | ab |
| ba | aa |
| cc | cc |
| ccc | |
| a | |
| b | |
| a | |
| a | |
| c | |



4☀. Số 3 ước

Cho dãy số nguyên dương A_1, A_2, \dots, A_N . Đếm xem trong dãy có bao nhiêu số hạng chỉ có đúng 3 ước dương.

Dữ liệu cho trong file **ThreeU.Inp** gồm:

- Dòng đầu ghi số nguyên dương N ($N \leq 2 \times 10^5$).
- Dòng thứ 2 ghi N số nguyên dương A_1, A_2, \dots, A_N ($A_i \leq 10^{12}$).

Kết quả ghi ra file **ThreeU.Out** là số các số hạng của dãy có đúng 3 ước.

Ví dụ:

| ThreeU.Inp | ThreeU.Out |
|-------------------|------------|
| 3 | 1 |
| 1 4 1000000000000 | |



5☀. Xếp công việc (Hard Version 2)

Có N công việc được đánh số thứ tự từ 1 đến N . Công việc thứ i có thời gian hoàn thành là A_i và hạn cuối cần hoàn thành là B_i . Nếu công việc thứ i được thực hiện xong trước hoặc ngay tại thời điểm B_i thì ta nói công việc i hoàn thành đúng thời hạn. Ngược lại công việc i không hoàn thành đúng thời hạn.

Yêu cầu: Hãy tìm cách sắp xếp các công việc sao cho số công việc thực hiện đúng hạn là nhiều nhất. Công việc đầu tiên có thể thực hiện từ thời điểm 0. Không được thực hiện nhiều hơn một công việc tại một thời điểm, thực hiện xong công việc này rồi mới chuyển sang công việc khác. Thời điểm hoàn thành công việc này và thời điểm bắt đầu công việc khác có thể bằng nhau. Công việc đầu tiên có thể bắt đầu từ thời điểm 0.

Dữ liệu cho trong file **WorkMax.Inp** gồm:

- Dòng 1 ghi số nguyên dương T ($T \leq 10$) là số bộ dữ liệu:



Mỗi bộ dữ liệu có dạng:

- Dòng thứ nhất ghi số nguyên dương N là số công việc.
- N dòng sau, dòng thứ i ghi hai số nguyên A_i và B_i .

Kết quả ghi ra file **WorkMax.Out** gồm T dòng. Mỗi dòng ghi số công việc nhiều nhất có thể thực hiện đúng hạn với bộ dữ liệu tương ứng.

Ví dụ:

| WorkMax.Inp | WorkMax.Out |
|-------------|-------------|
| 2 | 2 |
| 4 | 1 |
| 2 2 | |
| 3 3 | |
| 3 4 | |
| 8 10 | |
| 2 | |
| 2 2 | |
| 2 3 | |

Giới hạn:

- $3 \leq N \leq 10^5$;
- $1 \leq A_i \leq B_i \leq 10^9$.