

TỔNG QUAN ĐỀ THI

Bài	Tên bài	Tên file chương trình	Hạn chế thời gian	Hạn chế bộ nhớ	Điểm
1	Chuỗi online bạn bè	STREAK.*	1 giây	1024 MB	2
2	Hành trình xe điện	EVTRIP.*	1 giây	1024 MB	2.5
3	Trò chơi xếp chữ	WORDGAME.*	1 giây	1024 MB	2.5
4	Tối ưu giao dịch Blockchain	BLOCKOPT.*	1 giây	1024 MB	3

Dấu * được thay thế bởi PAS, CPP hoặc PY theo ngôn ngữ lập trình được sử dụng tương ứng là PASCAL, C++ hoặc Python.

LẬP TRÌNH GIẢI CÁC BÀI TOÁN SAU:

Bài 1: CHUỖI ONLINE BẠN BÈ (STREAK.*)

An sử dụng một ứng dụng nhắn tin và muốn biết bạn của mình, Bình, có những khoảng thời gian online liên tục dài nhất là bao lâu trong một ngày. Hệ thống ghi lại trạng thái của Bình mỗi phút trong suốt T phút của một ngày. Trạng thái có thể là một trong ba loại: "ONLINE", "IDLE" (không hoạt động), hoặc "OFFLINE".

Một "chuỗi online" được định nghĩa là một khoảng thời gian liên tục mà trạng thái của Bình là "ONLINE".

Yêu cầu: Cho một chuỗi các trạng thái của Bình trong T phút, hãy tìm độ dài của chuỗi online liên tục dài nhất. Nếu Bình không online phút nào, kết quả là 0.

Dữ liệu: Vào từ file văn bản STREAK.INP:

- Dòng đầu tiên chứa số tự nhiên T ($1 \leq T \leq 1440$), tổng số phút theo dõi trong ngày.
- T dòng tiếp theo, mỗi dòng chứa một xâu ký tự là trạng thái của Bình tại phút tương ứng: "ONLINE", "IDLE", hoặc "OFFLINE".

Kết quả: Ghi ra file văn bản STREAK.OUT một số nguyên duy nhất là độ dài của chuỗi online liên tục dài nhất.

Ví dụ:

STREAK.INP	STREAK.OUT
10 ONLINE ONLINE IDLE ONLINE ONLINE ONLINE OFFLINE ONLINE ONLINE IDLE	3
5 ONLINE ONLINE ONLINE ONLINE ONLINE	5
4 OFFLINE IDLE OFFLINE IDLE	0

Bài 2: HÀNH TRÌNH XE ĐIỆN (EVTRIP.*)

Một chiếc xe điện bắt đầu hành trình từ điểm 0 km với pin đầy, dung lượng pin tối đa là P_{\max} đơn vị. Mỗi km di chuyển tiêu tốn 1 đơn vị pin.

Trên quãng đường có N trạm sạc. Trạm sạc thứ i (với $i=1,2,\dots,N$) nằm ở vị trí D_i km tính từ điểm xuất phát và tại đó xe có thể sạc đầy pin (lên P_{\max}) ngay lập tức.

Xe không thể di chuyển nếu pin không đủ cho 1 km tiếp theo. Đích đến là thành phố ở vị trí D_{target} km.

Yêu cầu: Tìm số lần sạc ít nhất để xe có thể đi từ điểm 0 đến D_{target} . Nếu không thể đến đích, ghi ra -1.

Dữ liệu vào: File EVTRIP.INP

- Dòng đầu tiên: ba số tự nhiên $N, P_{\max}, D_{\text{target}}$ ($0 \leq N \leq 1000, 1 \leq P_{\max} \leq 10^9, 1 \leq D_{\text{target}} \leq 10^9$)
- N dòng tiếp theo: mỗi dòng chứa một số nguyên D_i ($1 \leq D_i < D_{\text{target}}$)

Kết quả: File EVTRIP.OUT

- Một số nguyên là số lần sạc ít nhất, hoặc -1 nếu không thể đến đích.

Input	Output
3 175 350 80 180 280	2
1 10 100 50	-1
1 50 75 30	1

Bài 3: TRÒ CHƠI XẾP CHỮ (WORDGAME.*)

Một trò chơi xếp chữ trên điện thoại, người chơi có một chuỗi ký tự s . Mỗi lượt chơi được phép xóa một ký tự bất kỳ. Trò chơi kết thúc khi chuỗi còn lại là palindrome (đọc xuôi ngược như nhau). Ví dụ: chuỗi “aca” hoặc “racecar” là chuỗi palindrome.

Yêu cầu: Tìm số lượt chơi ít nhất để chuỗi ban đầu trở thành palindrome.

Dữ liệu vào: WORDGAME.INP: chuỗi ký tự s ($1 \leq \text{độ dài} \leq 2000$, chỉ gồm chữ thường).

Dữ liệu ra: WORDGAME.OUT: số lượt chơi ít nhất.

Ví dụ minh họa chi tiết:

WORDGAME.INP	WORDGAME.OUT
abcca	1
abacda	3
racecar	0
abcde	4
ababa	0
programming	7

Bài 4: TỐI ƯU GIAO DỊCH BLOCKCHAIN (BLOCKOPT.*)

Trong một hệ thống blockchain, thợ đào chọn các giao dịch đang chờ (giả sử có N giao dịch) để đưa vào một khối mới. Mỗi giao dịch thứ i có phí F_i và kích thước S_i (với $i=1,2,\dots,N$) và F_i, S_i là các số nguyên dương. Khối có kích thước tối đa S_{\max} (S_{\max} là số nguyên dương). Thợ đào muốn tối đa tổng phí F_i sao cho tổng kích thước S_i không vượt S_{\max} .

Giả sử rằng có D ràng buộc phụ thuộc: nếu giao dịch A được chọn thì giao dịch B cũng phải được chọn. Các phụ thuộc này không tạo thành chu trình.

Yêu cầu: Tìm tổng phí giao dịch lớn nhất.

Dữ liệu: Vào từ file văn bản BLOCKOPT.INP:

- Dòng 1: N, S_{\max} ($1 \leq N \leq 50, 1 \leq S_{\max} \leq 1000$).
- N dòng tiếp: F_i, S_i ($1 \leq F_i, S_i \leq 1000$) cho giao dịch i .

- Dòng tiếp: D ($0 \leq D \leq \frac{N(N-1)}{2}$).
- D dòng tiếp (nếu $D > 0$): A, B (thể hiện cho ràng buộc nếu chọn A phải chọn B , trong đó A, B là chỉ số các giao dịch đang chờ $A, B \in \{1, 2, \dots, N\}$).

Kết quả: Ghi ra file văn bản BLOCKOPT.OUT tổng phí lớn nhất.

Ví dụ:

Input (BLOCKOPT.INP)	Output (BLOCKOPT.OUT)
3 10 10 5 7 4 3 3 1 1 2	17
3 10 10 5 10 5 5 3 0	20
2 10 100 11 200 12 0	0

-----HẾT-----

- Thí sinh không được sử dụng tài liệu;
- Giám thị không giải thích gì thêm.