

TỔNG QUAN BÀI THI

	Tên bài	File chương trình	Dữ liệu vào	Dữ liệu ra	Điểm
Bài 1	Phủ đoạn thẳng	Cover.*	Cover.inp	Cover.out	6
Bài 2	Uống nước	Water.*	Water.INP	Water.OUT	7
Bài 3	Phân tích số	Analyse.*	Analyse.INP	Analyse.OUT	7
Phân mở rộng . * là: .pas đối với NNLT Pascal; .cpp đối với NNLT C++					

Bài 1. Phủ đoạn thẳng

Cho một tập N đoạn thẳng (đánh số từ 1 đến N) với các đầu mút có tọa độ nguyên $[L_i, R_i], i = 1, 2, \dots, N$ và một đoạn thẳng $[P, Q]$.

Yêu cầu: Tìm một số ít nhất đoạn thẳng trong tập đã cho phủ kín đoạn $[P, Q]$ (tức là mỗi điểm $x \in [P, Q]$, x phải thuộc vào ít nhất một trong số các đoạn được chọn).

Dữ liệu vào cho trong tệp văn bản **COVER.INP**

- Dòng đầu tiên chứa 3 số nguyên N, P và Q ($N \leq 10^5, 0 < P \leq Q \leq 10^9$)
- N dòng tiếp theo mỗi dòng chứa hai số L_i, R_i ($0 < L_i \leq R_i \leq 10^9$)

Kết quả đưa ra tệp văn bản **COVER.OUT**

- Dòng 1: Đưa ra số nguyên K là số đoạn ít nhất chọn được hoặc đưa ra -1 nếu không chọn được các đoạn thỏa mãn.
- Dòng 2: Ghi K số thể hiện số hiệu của các đoạn thẳng được chọn.

Ví dụ:

COVER.INP	COVER.OUT
3 4 6	1
1 2	3
3 5	
4 7	

Subtasks:

- Có 60% số test ứng với 60% số điểm của bài có $N \leq 10^4, 0 < L_i \leq R_i \leq 10^5$
- Có 40% số test ứng với 40% số điểm của bài có $N \leq 10^5, 0 < L_i \leq R_i \leq 10^9$

Bài 2. Uống nước

An có N cái cốc được gán nhãn lần lượt $1, 2, \dots, N$ với dung tích không hạn chế và trong mỗi cốc đều có nước. An muốn uống hết số nước ở tất cả các cốc nhưng chỉ muốn uống ở đúng K cốc. Để có thể làm được như vậy, cậu ta phải rót nước từ cốc này sang cốc khác để chỉ còn đúng K cốc có nước.

Vấn đề đặt ra với An là chọn cốc nào để rót nước sang cốc nào, bởi vì khoảng cách giữa các cốc là không bằng nhau. Cụ thể, An sẽ mất một lượng sức khi rót nước từ cốc i sang cốc j gọi nó là C_{ij} .

Yêu cầu: Giúp An tìm thứ tự rót nước từ cốc này sang cốc khác để tổng lượng sức cậu ấy bỏ ra là ít nhất. Thỏa mãn điều kiện, An có thể uống hết lượng nước chỉ trong K cốc.

Input: vào cho trong tệp văn bản **WATER.INP**

- Dòng 1 chứa hai số nguyên N, K ($1 \leq K \leq N \leq 20$)
- N dòng sau, dòng thứ i chứa N số nguyên C_{ij} ($0 \leq C_{ij} \leq 10^5$) là lượng sức cần dùng khi rót nước từ cốc i sang cốc j . Nếu $i = j$ thì $C_{ij} = 0$.

Output đưa ra tệp văn bản **WATER.OUT** một số duy nhất là tổng lượng sức ít nhất cần sử dụng.

Ví dụ:

WATER.INP	WATER.OUT
5 2	5
0 5 4 3 2	
7 0 4 4 4	
3 3 0 1 2	
4 3 1 0 5	
4 5 5 5 0	

(Giải thích ví dụ: rót nước từ cốc 4->3 (tốn 1), sau đó rót từ 3->5 (tốn 2), rót từ 1->5 (tốn 2). Tổng ít nhất 5).

Subtasks:

- Có 40% số test ứng với 40% số điểm của bài có $1 \leq N \leq 10$
- Có 60% số test ứng với 60% số điểm của bài có $1 \leq N \leq 20$

Bài 3: Phân tích số

Cho một số nguyên dương S , S có thể phân tích thành dãy các số nguyên dương x_1, x_2, \dots, x_n sao cho $x_1 + x_2 + \dots + x_n = S$. Bài toán đặt ra là phân tích S thành tổng các số nguyên dương sao cho ước chung lớn nhất của các hạng tử này là 1. Cụ thể, cần tính số lượng dãy x_1, x_2, \dots, x_n sao cho $x_1 + x_2 + \dots + x_n = S$ và $\text{GCD}(x_1, x_2, \dots, x_n) = 1$ (trong đó, GCD là hàm ước chung lớn nhất). ví dụ $S=4$ ta có các dãy sau $\{1, 1, 1, 1\}$, $\{2, 1, 1\}$, $\{1, 1, 2\}$, $\{1, 2, 1\}$, $\{1, 3\}$, $\{3, 1\}$.

Yêu cầu: Cho số nguyên dương S tính số cách phân tích S thành tổng các số nguyên dương sao cho ước chung lớn nhất giữa các số là 1.

Input: file **Analyse.inp** có cấu trúc như sau:

Dòng đầu tiên chứa số nguyên dương t ($t \leq 10$) là số bộ test.

Tiếp theo là t dòng, mỗi dòng chứa một số nguyên dương s ($s \leq 10^9$).

Output: ghi kết quả vào **file Analyse.out**

Gồm t dòng, mỗi dòng là kết quả tương ứng mod module $10^9 + 7$.

Ví dụ

Analyse.inp	Analyse.out
2	1
1	6
4	

Giải thích

Trường hợp thứ nhất, chỉ có dãy $\{1\}$ thoả mãn.

Trường hợp thứ hai, có 6 dãy thoả mãn: $\{1, 1, 1, 1\}$, $\{1, 1, 2\}$, $\{1, 2, 1\}$, $\{1, 3\}$, $\{2, 1, 1\}$, $\{3, 1\}$.

Subtask:

- 30% test tương ứng với 30% số điểm có $S \leq 10$.
- 70% test tương ứng với 30% số điểm có $S \leq 10^9$.

----- Hết -----