

ĐỀ ĐỀ XUẤT

TỔNG QUAN ĐỀ THI

	Tên bài	Tên chương trình	File dữ liệu	File kết quả
Bài 1	Dãy số	DS.PAS	DS.INP	DS.OUT
Bài 2	Chi phí nhỏ nhất	TONGMIN.PAS	TONGMIN.INP	TONGMIN.OUT
Bài 3	Bãi đỗ xe	PARKING.PAS	PARKING.INP	PARKING.OUT

Bài 1. DÃY SỐ (7 điểm)

Vừa mới học xong kiến thức tính chất chia hết. Tuấn và Minh muốn cùng nhau kiểm tra các kiến thức vừa học. Tuấn viết một dãy số gồm n số nguyên không âm $a_1, a_2, \dots, a_n$ sau đó để Minh có bao nhiêu số trong dãy chia hết cho n . Vấn đề thật đơn giản, lần lượt thử kiểm tra từng số một với n . Sau khi đếm xong số lượng số chia hết cho n , Minh nhận thấy bài toán hơi nhàm chán liền nảy ra một ý, một bài toán vẫn có thể kiểm tra về kiến thức tính chia hết mà không tầm thường, đó là: đếm số lượng dãy con liên tiếp có tổng chia hết cho n rồi để Tuấn.

Yêu cầu: Cho n và dãy số nguyên không âm $a_1, a_2, \dots, a_n$, hãy đếm số lượng dãy con liên tiếp có tổng chia hết cho n .

Dữ liệu vào: Từ tệp văn bản **DS.INP** gồm:

- Dòng 1: chứa số nguyên n ($n \leq 10^5$)
- n dòng sau (từ dòng 2 đến dòng $n + 1$), mỗi dòng chứa một số nguyên không âm a_i ($a_i \leq 10^{50}$)

Kết quả: Ghi ra tệp văn bản **DS.OUT** gồm một số duy nhất là số lượng dãy con đếm được.

Ví dụ:

DS.INP	DS.OUT
3	3
3	
2	
1	

Bài 2 CHI PHÍ NHỎ NHẤT (7 điểm)

Giám đốc công ty ABC cần giải quyết M công việc của công ty (các công việc được đánh số từ 1 tới M). Hiện tại có N người thợ có thể thuê để làm cùng một lúc (các người thợ được đánh số từ 1 đến N). Với công việc thứ i , nếu thuê người thợ thứ j làm thì chi phí phải trả là $a_{i,j}$.

Yêu cầu: Hãy tìm cách thuê các người thợ sao cho số công việc được giải quyết là nhiều nhất và chi phí phải trả là nhỏ nhất. Biết rằng mỗi người thợ chỉ làm đúng một công việc.

Dữ liệu vào: Từ file văn bản **TONGMIN.INP** có cấu trúc:

- Dòng đầu ghi 2 số nguyên M, N. ($1 \leq M, N \leq 30$);
- Trong M dòng tiếp theo, dòng thứ i ghi N số nguyên dương $a_{i,j}$ với $i=1..M, j=1..N$. ($a_{i,j} \leq 32000$)

Kết quả: Ghi ra file văn bản **TONGMIN.OUT** gồm một số duy nhất là tổng chi phí phải trả.

Ví dụ:

TONGMIN.INP	TONGMIN.OUT
7 4	12
8 6 18 10	
15 1 15 2	
5 1 9 13	
11 14 12 12	
19 3 14 12	
20 4 2 17	
10 8 5 20	

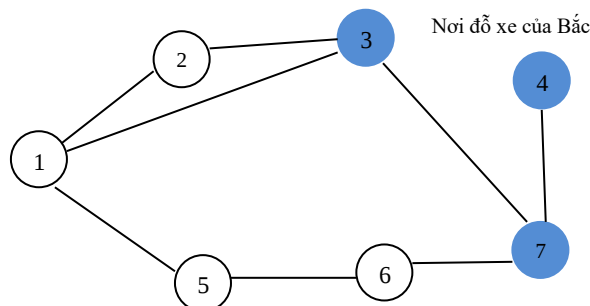
Bài 3. BÃI ĐỖ XE (6 điểm)

Khu tập thể nơi Bắc sống có một bãi đỗ xe ô tô gồm có N ($2 \leq N \leq 10^4$) khoang đỗ xe. Mỗi cặp khoang hoặc có đường đi trực tiếp hoặc có thể đi qua các khoang trung gian. Lối ra vào của bãi đỗ xe được đặt tại khoang 1, tất nhiên sẽ không thể có xe ô tô nào được đỗ ở khoang này.

Mỗi người muốn lái xe của mình ra khỏi bãi đỗ xe cần phải chọn một đường đi từ nơi đỗ xe của mình tới khoang 1, sao cho không có xe nào đỗ ở các khoang trên đường đi đó.

Hiện giờ trong bãi có K ($1 \leq K \leq N$) xe ô tô đang đỗ trong đó có xe của Bắc. Xe của Bắc đang ở khoang V ($1 \leq V \leq N$) và anh muốn lái xe ra khỏi bãi. Tuy nhiên, các đường đi tới khoang 1 có thể có xe chắn vì vậy muốn ra khỏi bãi Bắc phải gọi điện thoại di động tới chủ nhân của các xe nhờ họ lái xe ra khỏi bãi trước hoặc có thể lái xe sang một khoang trống khác (giả sử Bắc có tất cả các số điện thoại của chủ xe và cũng biết trước xe nào của ai). Bắc thì đang vội mà cước phí điện thoại thì đắt nên Bắc muốn chọn một con đường có ít xe chắn nhất để số cuộc gọi điện thoại là ít nhất.

Hình vẽ bên gồm có 7 khoang đỗ, trong đó 3 khoang đang có xe là khoang 3, 7 và 4. Hiện giờ xe của Bắc đang ở khoang 4, muốn lái xe ra ngoài cần phải gọi tối thiểu 1 cuộc điện thoại cho chủ xe đang ở khoang 7.



Yêu cầu: Bạn hãy lập trình để tính xem Bắc sẽ phải gọi ít nhất bao nhiêu cuộc điện thoại để nhờ các chủ xe.

Dữ liệu vào: Từ file **PARKING.INP**:

- Dòng 1 chứa các số N, M, K, V. Trong đó M là số đường đi trực tiếp giữa hai khoang đỗ xe ($1 \leq M \leq 10^5$)

- Dòng 2 là K số $P_1, P_2, \dots, P_K$ là chỉ số của của các khoang đang có xe đỗ.

- M dòng tiếp theo chứa M cặp (i, j) biểu diễn có đường đi hai chiều trực tiếp từ khoang i đến j.

Kết quả: Ghi ra file **PARKING.OUT** một số duy nhất là số cuộc gọi điện thoại ít nhất.

Ví dụ:

PARKING.INP	PARKING.OUT
7 8 3 4	1
3 4 7	
1 2	
1 3	
1 5	
2 3	
7 3	
4 7	
5 6	
6 7	

* *Chú ý:* Có 60% số test ứng với $N \leq 1000$

----- Hết -----