

TỔNG QUAN ĐỀ THI

STT	Tên file nộp	Input	Output	Điểm
1	NUMBER.cpp	NUMBER.inp	NUMBER.out	40
2	SEQ.cpp	SEQ.inp	SEQ.out	60
3	FLIGHT.cpp	FLIGHT.inp	FLIGHT.out	60
4	BRIDGES.cpp	BRIDGES.inp	BRIDGES.out	40

Bài 1: Số nguyên

Cho số nguyên dương A và các truy vấn sau:

1 c x: In ra “prime” nếu số A c x (c là các dấu ‘+’, ‘-’) là số nguyên tố. Nếu không, in ra “normal number”.

2 c x: In ra “cube number” nếu số A c x (c là các dấu ‘+’, ‘-’, ‘x’) là số lập phương. Nếu không, in ra “normal number”.

Input: File NUMBER.inp

- Dòng thứ nhất chứa hai số nguyên A và T ($1 \leq A \leq 10^6$).
- T dòng sau, mỗi dòng là 1 trong 2 truy vấn trên.

Output: File NUMBER.out

- In ra T dòng, mỗi dòng là kết quả của truy vấn tương ứng.

NUMBER.INP	NUMBER.OUT
5 2	normal number
1 + 9	normal number
2 x 2	

Ràng buộc:

#Subtask	Giới hạn		Số điểm
	T	Truy vấn	
0	Test mẫu		0
1	$T < 10^2$	$x = 0$	25
2	$T < 5 \cdot 10^5$	$0 < x < 10^6$	15

Bài 2: Dãy số

Cho dãy số $a[1], a[2], \dots, a[n]$ có n phần tử. Hãy:

- Kiểm tra xem số $a[i]$ có phải là số Fibonacci không. Nếu có, in ra 1, không, in ra -1.
- Cho T truy vấn, với mỗi truy vấn in ra chỉ số của phần tử đầu tiên lớn hơn X trong dãy.

Input: File SEQ.inp

- Dòng đầu tiên chứa một số nguyên n - số lượng các phần tử trong dãy.
- Dòng tiếp theo chứa n số nguyên dương $a[1], a[2], \dots, a[n]$.
- Dòng thứ ba chứa một số nguyên dương T - số lượng truy vấn.
- Dòng tiếp theo chứa T số nguyên dương X ($X \leq 10^{18}$).

Output: File SEQ.out

- In ra $T + 1$ dòng:
 - + Dòng đầu tiên chứa n số nguyên 1 hoặc -1.
 - + T dòng sau là kết quả của T truy vấn. Nếu không tìm thấy, in ra -1

SEQ.INP	SEQ.OUT
6	1 -1 1 1 -1 -1
5 9 8 2 7 6	1
5	2
5 8 1 3 9	1
	1
	-1

Ràng buộc:

#Subtask	Giới hạn				Số điểm
	N	$a[i]$	T	Khác	
0	Test mẫu				0
1	1000	$a[i] \leq 10^4$	100	$X \leq \max(a[i])$	15
2	1000	$a[i] \leq 10^6$	1000		25
3	10000	$a[i] \leq 10^{18}$	1000		12
4	100000	$a[i] \leq 10^{18}$	100000		8

Bài 3: Hủy chuyến bay

Arkady đã mua một chiếc vé máy bay từ thành phố A đến thành phố C. Thật không may, không có chuyến bay trực tiếp từ A đến C nhưng lại có nhiều chuyến từ A đến B và từ B đến C.

Có n chuyến bay từ A đến B, xuất phát lần lượt tại thời điểm $a[1], a[2], \dots, a[n]$ và đến B sau t_a phút.

Có m chuyến bay từ B đến C, xuất phát lần lượt tại thời điểm $b[1], b[2], \dots, b[m]$ và đến C sau t_b phút.

Thời gian chờ nối chuyến là không đáng kể. Do đó, Arkady có thể đi chuyến thứ i từ A đến B và chuyến thứ j từ B đến C khi và chỉ khi $b[j] \geq a[i] + t_a$.

Bạn có thể hủy tối đa k chuyến bay. Arkady không thể lên chuyến bay mà bạn đã hủy.

Arkady muốn đến C nhanh nhất có thể, còn bạn muốn anh ta đến C càng chậm càng tốt. Hãy tìm thời điểm sớm nhất mà Arkady có thể đến C nếu bạn hủy tối đa k chuyến bay tối ưu nhất. Nếu như sau khi bạn hủy, Arkady không thể đến C được, in ra -1.

Input: File FLIGHT.inp

- Dòng đầu tiên chứa 5 số nguyên n, m, t_a, t_b, k . ($k \leq n + m$; $0 < t_a \leq 10^5$; $0 < t_b \leq 10^5$)
- Dòng thứ hai chứa n số nguyên theo thứ tự tăng dần $a[1], a[2], \dots, a[n]$ ($a[i] \leq 10^9$).
- Dòng thứ ba chứa m số nguyên theo thứ tự tăng dần $b[1], b[2], \dots, b[m]$ ($b[i] \leq 10^9$).

Output: File FLIGHT.out

Nếu bạn có thể hủy tối đa k chuyến sao cho không còn cách nào để đi đến C, in ra -1.

Còn lại, in ra thời gian sớm nhất Arkady có thể đến C nếu bạn hủy tối đa k chuyến sao cho thời gian sớm nhất đến C là lớn nhất.

FLIGHT.INP	FLIGHT.OUT
4 5 1 1 2 1 3 5 7 1 2 3 9 10	11
2 2 4 4 2 1 10 10 20	-1

4 3 2 3 1 1 999999998 999999999 1000000000 3 4 1000000000	1000000003
---	------------

Giải thích:

Xét ví dụ đầu tiên, các chuyến bay xuất phát từ A lần lượt tại thời điểm 1, 3, 5, 7 và đến B tại 2, 4, 6, 8. Các chuyến bay xuất phát từ B lần lượt tại 1, 2, 3, 9, 10 và đến C tại 2, 3, 4, 10, 11. Bạn có thể hủy tối đa 2 chuyến bay trong số 9 chuyến bay này. Đáp án tối ưu là hủy chuyến bay đầu tiên từ A đến B (xuất phát tại thời điểm 1) và chuyến bay thứ tư xuất phát từ B đến C (xuất phát tại thời điểm 9). Khi đó Arkady chọn chuyến bay thứ ba từ A đến B và chuyến bay thứ năm từ B đến C.

Tại ví dụ thứ hai, bạn hủy toàn bộ các chuyến bay từ A đến B.

Ràng buộc:

#Subtask	Giới hạn		Số điểm
	n	m	
0	Test mẫu		0
1	10	10	15
2	1000	1000	21
3	50000	50000	14
4	$2 \cdot 10^6$	$2 \cdot 10^6$	10

Bài 4: Bắc cầu.

Hai nước Alpha và Beta nằm ở hai bên bờ sông Omega, Alpha nằm ở bờ bắc và có M thành phố được đánh số từ 1 đến M, Beta nằm ở bờ nam và có N thành phố được đánh số từ 1 đến N (theo vị trí từ đông sang tây). Mỗi thành phố của nước này thường có quan hệ kết nghĩa với một số thành phố của nước kia. Để tăng cường tình hữu nghị, hai nước muốn xây các cây cầu bắc qua sông, mỗi cây cầu sẽ là nhịp cầu nối 2 thành phố kết nghĩa. Với yêu cầu là các cây cầu không được cắt nhau và mỗi thành phố chỉ là đầu cầu cho nhiều nhất là một cây cầu, hãy chỉ ra cách bắc cầu được nhiều cầu nhất.

Input: File BRIDGES.inp

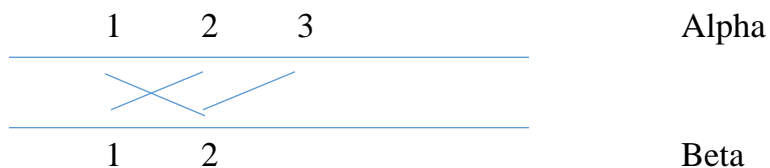
- Dòng đầu tiên chứa số ba nguyên dương M, N, T – số thành phố của đất nước Alpha, Beta và số cặp thành phố hữu nghị.
- T dòng tiếp theo chứa hai số nguyên a, b ($1 \leq a \leq M$, $1 \leq b \leq N$).

Output: File BRIDGES.out

- In ra một dòng chứa kết quả của bài toán

BRIDGES.INP	BRIDGES.OUT
3 2 3 1 2 2 1 3 2	2

Giải thích: Test mẫu trên có dạng như sau:



Ràng buộc:

#Subtask	Giới hạn		Số điểm
	M, N	T	
0	Test mẫu		0
1	$M, N \leq 10$	10	15
2	$M, N \leq 1000$	$T \leq N * M$, mỗi thành phố kết giao không quá 1 thành phố khác.	10
3	$M, N \leq 1000$	$T \leq N * M$	8
4	$M \leq 1000, N \leq 3 \cdot 10^5$	$T \leq N * M$	7

----- Hết -----

- Thí sinh không sử dụng tài liệu
- Thí sinh có thắc mắc vui lòng gửi câu hỏi trên trang Codeforces: <https://clctraining.contest.codeforces.com/> . BGK sẽ trả lời trong thời gian sớm nhất.