

Escape University

Disgraziatamente, precipiti nella prigione sotterranea segreta del Dipartimento di Matematica e Informatica di Ferrara. Dopo aver percorso un buio corridoio tappezzato con appunti di [Linguaggi Formali](#), [Calcolabilità e Complessità](#), raggiungi l'aula che manda sull'uscita (un portone chiuso, decorato da una grande scritta rossa).

Sullo schermo di un vecchio computer davanti a te è aperto il gestionale di un archivio. Puoi visionare il contenuto di N file. Ogni file n_i , $0 \leq i < N$, è identificato da un numero id_{n_i} e contiene una lista di numeri interi separati da uno spazio. Il primo, s_{n_i} , indica la dimensione del file, ovvero quanti interi contiene. Seguono $s_{n_i} - 1$ interi $x_{n_i}^0, x_{n_i}^1, \dots, x_{n_i}^{s_{n_i}-1}$. Schematizziamo la situazione con un esempio.

	File ($N = 5$)	Contenuto
n_0	8213	8 1 -3241 11 -84392 0 1 4
n_1	-2067	7 4 8 15 16 23 42
n_2	99	4 3 33 -3
n_3	23389	3 2 2147483647
n_4	42	1

Il file n_0 , ovvero il primo dei 5 file, è identificato dal numero $id_{n_0} = 8213$. Il suo primo intero, s_{n_0} , indica che il file contiene esattamente 8 interi, ovvero questo stesso e altri 7 indicati con $x_{n_0}^0, \dots, x_{n_0}^6$; in particolare, $x_{n_0}^0 = 1, x_{n_0}^1 = -3241, \dots, x_{n_0}^6 = 4$. Notiamo anche che non viene rispettato un ordinamento particolare, ci possono essere delle ripetizioni all'interno di uno stesso file (infatti $x_{n_0}^0 = x_{n_0}^5$) e un file potrebbe anche contenere la sua sola dimensione (si veda n_4).

Apprendo un sospettoso file README.md ti viene spiegato come aprire l'uscita per scappare dalla prigione. Devi scrivere un programma in C che legge Q interi, e per ognuno stampa il valore id del file che lo contiene. Se non è possibile rispondere, il programma deve stampare la stringa NULL. Nell'esempio sopra, la richiesta “-3241 33 5” deve restituire come risposta “8213 99 NULL” (vedi le sezioni **Input**, **Output** ed **Esempi** per più informazioni).

Note

Mentre un file può contenere lo stesso intero più volte, **non** esistono ripetizioni considerando file diversi. In altre parole, esiste una sola risposta corretta per ogni richiesta.

Attenzione: per risolvere questo problema, potrebbe essere necessario dover utilizzare un ordinamento. È obbligatorio utilizzare un algoritmo di ordinamento che si è scritto da soli.

Input

L'input deve essere letto da un file “input.txt”.

La prima riga dell'input consiste di due interi separati da uno spazio, N (il numero dei file presenti nell'archivio) e Q (il numero di richieste a cui rispondere).

Seguono N righe, ognuna delle quali descrive un file ed è così formattata: un intero id_{n_i} (il codice identificativo del file i -esimo n_i), un intero s_{n_i} (la dimensione del file n_i) e altri $s_{n_i} - 1$ interi (il resto del contenuto di n_i).

L'input termina con un'ultima riga contenente Q interi.

Output

L'output deve essere scritto su un file "output.txt".

Per ogni intero q letto nell'ultima riga dell'input, stampare in output la risposta al problema: l' id del file che contiene q oppure NULL se non esiste una risposta.

Vincoli

- $1 \leq N \leq 10^3$;
- $1 \leq Q \leq 10^6$;
- $-2^{31} \leq id, x < 2^{31}$;
- $1 \leq s \leq 10^3$;
- **tempo limite:** 3 s;
- **memoria limite:** 256 MiB.

Punteggio e casistiche

Risolvere questo problema garantisce 1/30 punti in più all'esame scritto finale. Ricorda: i punti sono cumulativi tra i vari esercizi, ed è possibile accumulare un massimo di 8/30 punti.

Ogni soluzione è testata considerando un certo numero di input, di difficoltà incrementale:

- 1: $1 \leq N, Q \leq 10^2$ (50 punti);
- 2: tutti i vincoli (50 punti);

Esempi

Input (input.txt)	Output (output.txt)
3 4 10 9 32 14 67 53 32 43 55 43 11 8 11 11 11 11 11 11 3 32 2 7 7 14 32 56	32 10 10 NULL
5 3 8213 8 1 -3241 11 -84392 0 1 4 -2067 7 4 8 15 16 23 42 99 4 3 33 -3 23389 3 2 2147483647 42 1 -3241 33 5	8213 99 NULL