

Problema Parcare

Fișier de intrare `parcare.in`
Fișier de ieșire `parcare.out`

În cel mai recent eveniment al companiei Tesla, Paul Musk a anunțat un nou produs inovativ: parcare autonomă. Fiind cunoscut pentru lansările produselor incomplete, nici parcare nu este completă, fiind nevoie de o automatizare pentru a atribui câte un loc mașinilor care vor să folosească parcare.

Parcare este formată din N locuri, numerotate de la 1 la N , și este deschisă timp de T secunde, începând cu secunda 1. Pe parcursul zilei, sosesc M mașini care vor să folosească parcare, pentru fiecare dintre acestea știindu-se timpul de sosire s_i și timpul de plecare p_i . Mașinile vin în ordinea timpului de sosire s_i și ocupă locul de parcare în intervalul de timp $[s_i, p_i]$. Pentru fiecare dintre acestea, trebuie să afișați un loc liber de parcare (dacă sunt mai multe, se poate afișa oricare) în care aceasta se poate așeza sau -1 dacă parcare este plină în momentul venirii mașinii. Dacă o mașină nu are loc în parcare la timpul de sosire, aceasta nu va mai intra în parcare la niciun timp viitor.

La final, Paul este interesat de mașinile care mai sunt rămase în parcare la închiderea parcarii, de aceea, vă cere să afișați configurația parcarii la timpul T .

Date Intrare

Pe prima linie se găsesc trei numere întregi N, M și T , reprezentând numărul de locuri din parcare, numărul de mașini care vin să folosească parcare, respectiv numărul de secunde pentru care este deschisă parcare.

Următoarele M linii conțin fiecare câte două numere întregi s_i, p_i , reprezentând venirea unei mașini la secunda s_i care va pleca la secunda p_i .

Mașinile apar în fișierul de intrare în ordine crescătoare după timpul de sosire s_i .

Date Ieșire

Se vor afișa $M + 1$ linii în total, primele M linii conținând fiecare câte un număr întreg între 1 și N reprezentând locul de parcare pe care îl va ocupa mașina, sau -1 dacă nu există niciun loc de parcare disponibil.

Ultima linie va conține N numere întregi, reprezentând configurația parcarii la închidere, unde cel de-al i -lea număr reprezintă **timpul de sosire** al mașinii de pe locul de parcare i , sau -1 dacă locul de parcare i este gol.

Restricții

- $1 \leq N, M, T \leq 200\,000$
- $1 \leq s_i \leq T$
- $1 \leq s_i < p_i \leq 200\,000$
- Considerând următoarele $2M$ valori: $s_1, s_2, \dots, s_M, p_1, p_2, \dots, p_M$, acestea sunt distincte două câte două.
- Dacă există mai multe soluții, se poate afișa oricare dintre acestea.**

#	Punctaj	Restricții
1	24	$s_i + 1 = p_i$, adică fiecare mașină stă exact o secundă.
2	26	$p_i > s_j$, adică toate mașinile vin înainte ca vreo mașină să plece.
3	26	$N \leq 1\,000$
4	24	Fără restricții suplimentare.

Exemple

parcare.in	parcare.out	Explicații
2 4 6	2	Prima mașină sosește în secunda 1 și este parcată pe locul 2.
1 3	1	A doua mașină sosește în secunda 2 și este parcată pe locul 1.
2 10	2	În secunda 3 se eliberează locul 2.
4 6	-1	Cea de-a treia mașină sosește în secunda 4 și ocupă locul 2.
5 8	2 -1	Mașina sosită în secunda 5 nu găsește loc liber.
		În secunda 6 se eliberează locul 2.
		După închiderea parcarii, pe locul 1 va fi parcată mașina venită în secunda 2, locul al doilea fiind liber.