

## Problema CLO

### Nori

Day 1, fișiere sursă clo.\*

Memorie disponibilă: 64 MB. Timp maxim de execuție: 3 s.



Pe cer sunt  $n$  nori, toți deplasându-se în mod continuu în aceeași direcție, cu aceeași viteză constantă  $v=(v_x, v_y)$ . De exemplu: pentru orice număr real  $t \geq 0$  și orice punct al unui nor cu coordonate inițiale  $(x,y)$ , poziția acestui punct după timpul  $t$  este  $(x+t*v_x, y+t*v_y)$ .

Pentru simplitate, vom presupune că fiecare nor este un poligon (care include frontiera), ale cărui vârfuri au coordonatele întregi. Poligoanele nu trebuie să fie neapărat convexe, dar laturile nu se autointersectează (cu excepția celor care au capete comune ale laturilor consecutive). Norii se pot intersecta.

Pe pământ se află un centru de control al sateliților, care are coordonatele  $(0,0)$  și exact deasupra lui și al norilor se află un satelit. Pe verticală din centrul de control se trimite o rază laser. Raza laser este folosită pentru a comunica cu satelitul. Dar, dacă raza laser intersectează un nor, atunci comunicarea cu satelitul nu mai este posibilă. Inițial raza laser, nu intersectează nici un nor. În timpul observațiilor, pot exista mai multe momente când raza laser intersectează nori (unul sau mai mulți) întrerupând comunicarea cu satelitul. Chiar și atunci când raza laser intersectează vârful unui nor, comunicarea este întreruptă. Trebuie să scrieți un program care calculează de câte ori comunicarea este întreruptă, până când norii depășesc raza laser.

### Cerință

Scrieți un program, care:

- citește de la intrarea standard pozițiile, formele și viteza norilor
- determină de câte ori comunicarea este întreruptă
- scrie rezultatele la ieșirea standard

### Date de intrare

Prima linie conține trei numere întregi  $n$ ,  $v_x$ , și  $v_y$ , separate printr-un singur spațiu,  $1 \leq n \leq 1000$ ,  $-1\,000\,000\,000 \leq v_x, v_y \leq 1\,000\,000\,000$ ,  $n$  este numărul de nori,  $v=(v_x, v_y)$  este vectorul vitezei al norilor ( $v \neq (0,0)$ ). Abscisele corespund direcției Vest – Est, iar ordonatele corespund direcției Nord – Sud. Următoarele  $n$  linii conțin date despre nori, câte un nor pe o linie. Fiecare dintre aceste linii conține un șir de numere întregi, separate printr-un singur spațiu. Primul număr întreg  $k$ , de pe o astfel de linie reprezintă numărul de vârfuri ale unui nor,  $3 \leq k \leq 1000$ . Urmează  $2k$  numere întregi:  $x_1, y_1, x_2, y_2, \dots, x_k, y_k$ ,  $-1000000000 \leq x_i, y_i \leq 1000000000$ ;  $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$  sunt coordonatele vârfurilor consecutive ale unui nor considerate în sensul acelor de ceasornic.

### Date de ieșire

Prima și singura linie va conține exact un număr întreg, care reprezintă de câte ori comunicarea este întreruptă.

### Exemplu:

pentru datele de intrare:

```
4 -2 -1
4 6 2 6 4 8 4 8 2
4 2 3 1 -1 2 5 4 2
3 -3 1 -1 2 -1 -1
5 5 3 3 3 3 5 6 5 6 -1
```

rezultatul corect este:

3

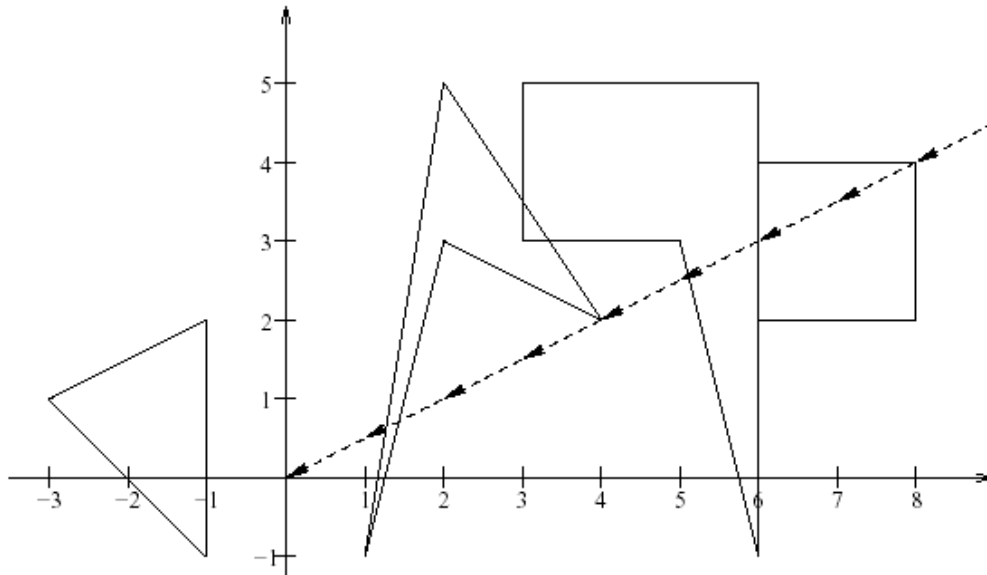


Figura reprezintă norii văzuți de jos. Linia întreruptă marchează punctele pe care le va traversa raza laser.

## Problema JOU

### Călătorie

Day 0, fișiere sursă jou.\*

Memorie disponibilă: 32 MB. Timp maxim de execuție: 0.5 s.



În Byteland există  $n$  orașe (numerotate prin  $1, 2, \dots, n$ ), legate între ele prin șosele bidirecționale. Regele din Byteland nu este foarte generos, așa încât nu există decât  $n-1$  șosele, dar ele leagă orașele în așa fel încât este posibil să călătorim între orice două orașe.

Într-o zi, un turist din Byteland ajunge în orașul  $k$ . Turistul vrea să călătorească pornind din orașul  $k$  vizitând în drumul său orașele  $m_1, m_2, \dots, m_j$  (nu neapărat în această ordine) – numerele  $m_i$  sunt distincte și diferite de  $k$ . Turistul are o anumită sumă limitată de bani și ar vrea să viziteze toate orașele planificate pe cel mai scurt traseu posibil (pornind din orașul  $k$ ). Un traseu este o secvență de una sau mai multe șosele, cu proprietatea că o șosea începe cu orașul în care a ajuns pe șoseaua anterioară din secvență. Ajutați turistul să determine lungimea celui mai scurt traseu din călătoria sa.

### Cerință

Scrieți un program, care:

- citește de la intrarea standard următoarele:
  - o descriere a drumurilor care leagă orașele din Byteland
  - numărul orașului de unde turistul începe călătoria
  - o listă de orașe pe care turistul vrea să le viziteze
- determină lungimea minimă călătoriei turistului
- scrie rezultatele la ieșirea standard

### Date de intrare

Prima linie conține două numere întregi  $n$  și  $k$  separate printr-un singur spațiu ( $2 \leq n \leq 50000$ ,  $1 \leq k \leq n$ ),  $n$  este numărul de orașe din Byteland și  $k$  este numărul orașului din care turistul își începe călătoria. Fiecare din următoarele  $n-1$  linii conține o descriere a unui drum din Byteland. A  $i+1$ -a linie (pentru  $1 \leq i \leq n-1$ ) conține trei numere întregi  $a_i$ ,  $b_i$  și  $d_i$  separate prin câte un spațiu ( $1 \leq a_i, b_i \leq n$ ,  $1 \leq d_i \leq 1000$ ),  $a_i$  și  $b_i$  sunt orașe legate printr-o șosea de lungime  $d_i$ . A  $n+1$ -a linie conține un număr întreg  $j$  – numărul de orașe pe care turistul vrea să le viziteze ( $1 \leq j \leq n-1$ ). Următoarea linie conține,  $j$  numere întregi  $m_i$ , distincte, separate prin câte un singur spațiu – numerele asociate orașelor pe care turistul vrea să le viziteze ( $1 \leq m_i \leq n, m_i \neq k$ ).

### Date de ieșire

Prima și singura linie va conține exact un număr întreg: lungimea celui mai scurt traseu al călătoriei turistului.

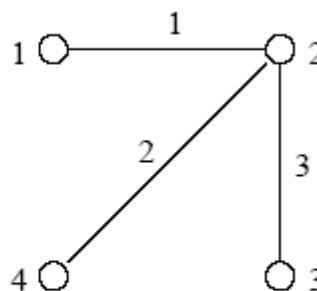
### Exemplu:

pentru datele de intrare

```
4 2
1 2 1
4 2 2
2 3 3
2
1 3
```

rezultatul corect este

5





## Problema SWE

### Bomboane

Day 1, fișiere sursă swe.\*

Memorie disponibilă: 64 MB. Timp maxim de rulare: 0.1 s

John are  $n$  borcane cu bomboane. Fiecare borcan conține diferite bomboane. Al  $i$ -lea borcan conține  $m_i$  bomboane. John s-a hotărât să mănânce o parte dintre bomboanele sale. Ar vrea să mănânce cel puțin  $a$  bomboane, dar nu mai mult decât  $b$  bomboane. Problema este că John nu se poate hotărâ cât de multe bomboane și de ce fel ar vrea să mănânce. În câte moduri poate el să facă acest lucru?

### Cerință

Scrieți un program, care:

- citește de la intrarea standard numărul de bomboane din fiecare borcan și valorile  $a$  și  $b$
- calculează numărul de posibilități pe care le are John pentru a manca bomboanele (respectând condițiile de mai sus)
- scrie rezultatul la ieșirea standard

### Date de intrare

Prima linie a fișierului de intrare conține trei numere întregi separate printr-un singur spațiu ( $1 \leq n \leq 10$ ,  $0 \leq a \leq b \leq 10000000$ ). Fiecare din următoarele  $n$  linii conține câte un număr întreg. Linia  $i+1$  conține întregul  $m_i$  -numărul de bomboane din borcanul cu numărul  $i$  ( $0 \leq m_i \leq 10000000$ ).

### Date de ieșire

Fie  $k$  numărul de moduri diferite în care John poate să-și aleagă bomboanele. Prima și singura linie de la ieșirea standard va conține un număr întreg:  $k \bmod 2004$  (adică restul împărțirii lui  $k$  la 2004).

### Exemplu

Pentru datele de intrare:

```
2 1 3
3
5
```

rezultatul corect este:

```
9
```

care corespunde următoarelor posibilități de alegere:

```
(1,0) (2,0) (3,0) (0,1) (0,2) (0,3) (1,1) (1,2) (2,1)
```

## Problema TRI

### Excursii

Day 1, fișiere sursă tri.\*

Memorie disponibilă: 64 MB. Timp maxim de execuție: 3 s.



În legătură cu vacanța care urmează, o mulțime de oameni vor să plece în excursii de neuitat. Pentru a se distra, fiecare vrea să meargă în excursie cu un grup de prieteni. O agenție de turism oferă anumite excursii. Fiecare excursie se adresează unui grup de persoane, care trebuie să se încadreze între un număr minim și un număr maxim de participanți. Fiecare grup poate să-și aleagă o singură excursie. În plus, fiecare excursie poate fi aleasă de un singur grup. Agenția de turism vă cere ajutorul. Ea vrea să organizeze cât mai multe excursii. Sarcina voastră este să determine excursia pentru fiecare grup, astfel încât să se organizeze un număr maxim de excursii.

### Cerință

Scrieți un program, care:

- citește descrierea grupurilor și a excursiilor de la intrarea standard
- găsește excursia pentru fiecare grup, astfel încât să se organizeze un număr maxim de excursii.
- scrie rezultatul la ieșirea standard.

Dacă există mai multe soluții programul vostru va da o singură soluție.

### Date de intrare

Prima linie conține două numere întregi:  $n$  și  $m$  separate printr-un spațiu,  $1 \leq n \leq 400000$ ,  $1 \leq m \leq 400000$ ;  $n$  este numărul de grupuri și  $m$  este numărul de excursii. Grupurile sunt numerotate de la 1 la  $n$  și excursiile de la 1 la  $m$ .

Următoarele  $n$  linii conțin dimensiunile grupurilor, un număr pe o linie. A  $i+1$ -a linie conține numărul întreg  $s_i$  – dimensiunea grupului  $i$ ,  $1 \leq s_i \leq 10^9$ .

Următoarele  $m$  linii conțin descrierile excursiilor, o excursie pe o linie. Linia  $n+j+1$  conține două numere întregi  $l_j$  și  $u_j$ , separate printr-un singur spațiu.  $l_j$  și  $u_j$  sunt minimul, respectiv maximul dimensiuni grupului care poate merge în excursia respectivă,  $1 \leq l_j \leq u_j \leq 10^9$ .

### Date de ieșire

Prima linie va conține un număr întreg  $k \geq 0$  – numărul maxim de excursii care pot fi organizate. Următoarele  $k$  linii vor conține descrierile asocierilor grup – excursie. Fiecare dintre aceste linii va conține o pereche de numere întregi, separate printr-un singur spațiu: număr asociat grupului și număr asociat excursiei. Pot exista mai multe soluții, dar programul vostru va afișa una dintre ele.

### Exemplu

pentru datele de intrare:

```
5 4
54
6
9
42
15
6 6
20 50
2 8
7 20
```

rezultatul corect este:

```
3
2 1
3 4
4 2
```