

BARAJ

Problema 1

ajutor

Sursa : ajutor.c, ajutor.cpp sau ajutor.pas

Dacă te-ai rănit, nu folosi Sprite să tratezi rana. Cel mai bine e să fugi la cel mai apropiat post de prim ajutor. Norocul tău este că ai o hartă din care poți afla coordonatele carteziene ale posturilor de prim ajutor. Ghinionul este că, din cauza durerii, nu poți să fugi direct spre un post, ci numai pe direcțiile nord-sud și est-vest. Ca să știi dacă mai are sens să fugi spre un post sau să te bucuri de ultimele clipe de viață, cel mai bine e să evaluezi distanța până la cel mai apropiat post de prim ajutor.

Cel mai apropiat post este cel pentru care distanța Manhattan este minimă.

De data aceasta ai scăpat pentru că timpul necesar ajungerii până la post a fost suficient ; însă îți pui întrebarea: dacă accidentul s-ar fi întâmplat în alt loc, ai fi scăpat ?

Cerință:

Se consideră **N** puncte în plan, reprezentând posturile de prim ajutor și alte **M** puncte reprezentând posibile locații ale accidentului. Se cere pentru fiecare dintre cele **M** puncte distanța Manhattan până la cel mai apropiat post.

Date de intrare:

Fișierul **ajutor.in** conține pe prima linie numerele întregi **N** și **M**, în această ordine, separate printr-un spațiu. Pe următoarele **N** linii se află câte două numere întregi, separate printr-un spațiu, reprezentând coordonatele fiecărui post de prim ajutor. Pe următoarele **M** linii se află câte două numere întregi, separate printr-un spațiu, reprezentând coordonatele punctelor de accident. Coordonatele se dau în ordinea (abscisa, ordonata).

Date de ieșire:

Fișierul **ajutor.out** va conține **M** linii, cu câte un număr pe fiecare linie, reprezentând distanța minimă până la cel mai apropiat post de prim ajutor.

Restricții:

- $0 < N < 401$
- $0 < M < 500001$
- oricare coordonată este un număr întreg din intervalul $[0, 32000]$

Observații:

- Dacă te afli deja la un post de prim ajutor (coordonatele sunt identice) distanța e **0**.
- Distanța Manhattan este cel mai scurt drum între două puncte, mergând doar pe direcții paralele cu axele de coordonate, adică $|x_1 - x_2| + |y_1 - y_2|$, unde (x_1, y_1) , (x_2, y_2) sunt coordonatele celor 2 puncte.
- În fișierul de intrare pot exista puncte cu aceleași coordonate.
- Se acordă punctele pentru fiecare test, doar dacă toate valorile din fișierul de ieșire sunt corecte.
- Tot ce face Sprite e să-ți potolească setea.

Exemplu:

ajutor.in

```
4 4
1 1
5 5
1 5
5 1
0 0
1 1
3 3
4 1
```

ajutor.out

```
2
0
4
1
```

Timp maxim de execuție/test: 1.3 secunde pentru Linux și 2 secunde pentru Windows.

BARAJ

Problema 2

ghizi

Sursa : **ghizi.c**, **ghizi.cpp** sau **ghizi.pas**

Se caută ghizi pentru Olimpiada Națională de Informatică. Deoarece la Olimpiadă participă K echipe, trebuie ca în fiecare moment de timp din intervalul $[0, 100)$ să existe exact K ghizi.

Pentru posturile de ghid s-au înscris N voluntari, care au fost numerotați distinct de la 1 la N . Fiecare dintre cei N voluntari a specificat un interval de timp în care poate asigura servicii de ghid. Voluntarul i poate fi ghid în intervalul $[T1_i, T2_i)$ (intervalul este închis în $T1_i$ și deschis în $T2_i$).

Cerință:

Dându-se intervalele de timp asociate celor N voluntari, determinați o variantă de angajare astfel încât în fiecare moment de timp să fie prezenți **exact** K ghizi. Numărul total de voluntari angajați este irelevant.

Date de intrare:

Prima linie a fișierului de intrare **ghizi.in** conține două numere întregi N și K , separate printr-un spațiu, cu semnificațiile de mai sus. Voluntarii sunt numerotați distinct, de la 1 la N . Fiecare dintre următoarele N linii conține descrierea unui voluntar; mai exact linia $i+1$ conține valorile întregi $T1_i$ și $T2_i$ pentru voluntarul i .

Date de ieșire:

În fișierul **ghizi.out** veți afișa pe prima linie numărul total de ghizi angajați (M). Pe a doua linie veți scrie M numere distincte între 1 și N , ordonate crescător, reprezentând numerele de ordine ale ghizilor angajați.

Restricții și precizări:

- $1 \leq N \leq 5000$
- $0 \leq T1 < T2 \leq 100$ pentru fiecare dintre cei N voluntari
- $1 \leq K \leq N$
- Pot exista 2 voluntari cu același interval asociat.
- Toate testele date vor avea soluție.
- Dacă există mai multe soluții, afișați una oarecare.
- La preselecție participă numai fete.

Exemplu:

ghizi.in	ghizi.out
6 2	4
0 100	1 4 5 6
0 15	
15 99	
0 10	
10 20	
20 100	

Timp maxim de execuție: 0.1 secunde/test sub Linux și 0.3 secunde sub Windows.

BARAJ

Problema 3

sala

Sursa : sala.c, sala.cpp sau sala.pas

Președintele unui **Concurs de Informatică** dorește ca la festivitatea de încheiere să fie o atmosferă plăcută. Pentru acest lucru el vrea să așeze participanții pe n rânduri după anumite reguli. Aceste reguli sunt:

- pe primul rând al sălii așează una lângă alta n persoane.
- pe rândurile următoare așezarea pe poziția i (numerotarea se face de la stânga la dreapta) este condiționată de așezarea persoanelor de pe rândul anterior, adică dacă pe rândul din față pe pozițiile i și $i+1$ stau fie numai băieți fie numai fete, atunci se va așeza o fată, iar dacă pe aceste poziții stau persoane de sex opus se va așeza un băiat.

Conform acestei reguli pe rândul cu numărul de ordine i ($i \in \{1, 2, \dots, n\}$) se vor așeza $n-i+1$ persoane.

Cerință:

Pentru n dat se cere să se determine numărul maxim de băieți ce pot lua loc în sală, astfel încât să se respecte regulile anterioare.

Date de intrare:

În fișierul text **sala.in** pe prima linie se află numărul de cifre ale lui n , iar pe linia a doua se află cifrele numărului n separate între ele prin câte un spațiu.

Date de ieșire:

În fișierul text **sala.out** pe prima linie se va afișa numărul din cerință.

Restricții:

$$1 \leq n \leq 10^{101}$$

Pentru 30% din teste $n < 100$

Pentru 70% din teste, $n < 30000$

Exemplul 1:

sala.in

1

5

sala.out

10

Exemplul 2:

sala.in

2

1 3

sala.out

61

Timp de execuție/test: 0.1 secunde (atât sub Windows cât și sub Linux).