

Úlohy 1. kola letnej časti, kategória T

1. Trasy lodí

kat. T; 0 b za popis, 20 b za program

Nedaleko kalifornského pobrežia je ostrov. Má svoje meno, ale miestni mu hovoria Ostrov. Na ostrove je maják. V majáku žije jeho správca. Má svoje meno, ale miestni mu hovoria Maják.

Maják z majáku dovidí ľubovoľným smerom do tej istej vzdialenosti. Môže teda pozorovať, čo sa deje vo vnútri konkrétneho kruhu, ktorého stredom je samotný maják. Všade v tomto kruhu okrem samotného majáku je voda.

Kruhom sa raz za čas preplaví nejaká loď. Všetky lode sa plavia po priamke (z ktorej Maják vidí len úsečku). Maják si do databázy poctivo zapisoval, kedy, odkiaľ a kam sa ktorá loď plavila.

Občas sa stane, že sa okolo majáku plavia dve lode naraz. A v tých najvzrušujúcejších chvíľach to dokonca chvíľu vyzerá na kolíziu. Vtedy jedna z lodí musí spomaliť alebo zrýchliť. A Maják je celý preč od radosti, že sa konečne niečo zaujímavé deje.

Po poslednej oslave Majákových narodenín však s kamarátmi bľbli, menili heslá a z databázy zmazali stĺpec s časmi. Na nové heslá si ešte Maják ráno horko-ťažko spomenul, časy do databázy mu už ale nik nevráti.

Úloha

Na obvode kruhu, ktorý vidí Maják z majáku, rovnomerne rozmiestnime r bodov (kde r je nepárne). Idúc po obvode kruhu body očísľujeme od 0 po $r - 1$. Lodí sa doteraz okolo plavilo n , tie očísľujeme od 0 po $n - 1$. O poradí lodí nič nevieme – napr. loď 3 mohla prejsť okolo skôr ako loď 47, neskôr ako loď 47, alebo zhruba v tom istom čase. Pre každé i platí, že loď i Maják prvýkrát uvidel v bode z_i a naposledy v bode k_i .

Váš program dostane na vstupe vyššie popísané údaje. Z nich by mal vypočítať počet dvojíc lodí $i < j$ takých, že je možné, že sa lode i a j museli jedna druhej prispôsobiť aby zabránili kolízii.¹ Pochopiteľne, lode považujte pri riešení úlohy za body.

Formát vstupu

V prvom riadku je číslo n udávajúce počet lodí. V druhom riadku je číslo r udávajúce počet bodov na obvode kruhu (pričom r je nepárne číslo² väčšie ako $2n$).

Nasleduje n riadkov, každý z nich popisuje jednu loď: obsahuje jej čísla z_i a k_i . Platí $0 \leq z_i < r$ a $0 \leq k_i < r$. Navyše platí, že všetky hodnoty z_i a k_i sú navzájom rôzne. (Vstup teda obsahuje $2n$ rôznych súradníc.)

Formát výstupu

Vypíšte jeden riadok a v ňom jedno celé číslo: počet dvojíc lodí, ktoré sa mohli vo vnútri kruhu stretnúť.

Hodnotenie

Je päť sád vstupov.

Maximálne počty lodí v týchto sádach sú 20, 75 000, 200 000, 75 000 a 200 000.

Maximálne hodnoty r v týchto sádach sú 1000, 10^6 , 10^{18} , 10^{18} a 10^{18} .

Navyše v tretej sade platí, že všetky vstupy sú generované rovnomerne náhodne (každé z čísel z_i a k_i je zvolené náhodne spomedzi všetkých ešte nepoužitých).

¹Povšimnite si, že každú dvojicu treba posudzovať zvlášť. *Nepýtame* sa teda na to, koľko najviac takých dvojíc mohlo *postupne* nastať. Rozmyslite si sami, či a prečo je v tom rozdiel.

²To aby Majákovi nechodili lode cez maják.

Príklady

vstup

```
3
101
10 20
15 25
70 60
```

výstup

```
1
```

Lode 0 a 1 sa mohli stretnúť, ak sa plavili okolo majáku s Majákom zhruba v tom istom čase.

vstup

```
3
101
97 3
6 94
91 10
```

výstup

```
0
```

vstup

```
4
101
0 50
25 75
20 80
82 22
```

výstup

```
4
```

2. Treba vysávať

kat. T; 0 b za popis, 20 b za program

Georg sa po desaťročnej zaoceánskej plavbe vrátil do svojej vily a zistil, že po celých desať rokov v nej nikto nevysával. Tak sa do toho hneď pustil a plný energie začal vysávať, čo mohol. Ale po chvíli si všimol, že si to mal lepšie naplánovať – vo svojom zápale povysával niektoré miesta viackrát a niektoré vôbec. Teraz potrebuje vašu pomoc.

Úloha

Georgova vila má n miestností a $n - 1$ chodieb. Každá chodba spája práve dve miestnosti, a medzi každými dvoma miestnosťami vedie práve jedna cesta.

Georg postupne robí q rôznych činností – buď vysáva, alebo premýšľa. Vysáva tak, že začne v miestnosti a , ide do miestnosti b , a cestou povysáva každú chodbu, ktorou prejde. Premýšľa tak, že príde do nejakej chodby a snaží sa spomenúť si, kolkokrát ju už povysával. S tým mu musíte pomôcť.

Formát vstupu a výstupu

Na prvom riadku sú čísla n a q ($1 \leq n, q \leq 100\,000$) udávajúce počet miestností a počet činností. Miestnosti sú očíslované od 1 po n .

Nasleduje $n - 1$ riadkov, ktoré popisujú chodby. Každý obsahuje dve čísla p, q udávajúce, že miestnosti p a q sú spojené chodbou.

Ďalších q riadkov jednotlivé popisuje činnosti. Na každom sú tri čísla t, x, y . Aby sme zabezpečili, že činnosti naozaj vyhodnocujete v zadanom poradí, musíte ich najprv dekodovať: Nech v je posledné číslo, čo ste zatiaľ vypísali, alebo 0, ak ste zatiaľ nevypísali nič. Potom nech $a = x \text{ xor } v$, $b = y \text{ xor } v$.

Keď vypočítate a a b , číslo t vám povie typ činnosti, čo máte spraviť. Platí $1 \leq a, b \leq n$ a $a \neq b$. Ak je $t = 1$, Georg povysáva všetky chodby medzi miestnosťami a a b (nemusíte nič vypísať). Ak je $t = 2$, Georg príde do chodby medzi miestnosťami a a b (určite budú spojené chodbou) a chce, aby ste vypísali jedno číslo: počet, kolkokrát ju zatiaľ povysával. (Toto číslo bude nová hodnota v .)

Príklad

vstup

```
6 7
2 4
1 2
4 6
4 5
3 2
2 4 5
1 6 1
1 4 3
1 2 6
2 4 2
1 0 5
2 1 0
```

výstup

```
0
3
2
```

Na predposlednom riadku je $a = 3, b = 6$. Na poslednom riadku je $a = 2, b = 3$.

3. Tesná medzera v stene

kat. T; 0 b za popis, 20 b za program

Pytón Peľko (možno ste sa s nim stretli, keď ste riešili seminár PRASK) mal dosť nepríjemný deň. Nejaký záškodník mu do jeho obľúbenej ovsenej kaše, ktorú si dáva každé ráno, primiešal cement. Chudák Pytón je teraz plný stvrdnutého cementu a nedokáže sa ohýbať.

Ako sa teraz dostane domov?

Úloha

Pytóna Peľka si môžeme predstaviť ako lomenú čiaru v 2D. Celou čiarou môžeme hýbať aj ju otáčať (v 2D), ale nemôžeme meniť jej tvar.

Celý pytón je na jednej strane veľkej steny (priamky) a pytónove obydlie je na druhej strane steny. V stene je tenučká štrbina. Pre jednoduchosť si predstavme, že pytón je nekonečne tenký a štrbina je nekonečne úzka.

Zistite, či sa dokáže pytón prepchať cez štrbinu v stene a dostať sa domov.

Formát vstupu

Na prvom riadku vstupu je číslo n ($1 \leq n \leq 1000$) udávajúce počet vrcholov lomenej čiary. Nasleduje n riadkov a na i -tom z nich sú dve čísla x_i, y_i ($0 \leq x_i \leq 10^9, 1 \leq y_i \leq 10^9$) udávajúce pozíciu i -teho vrchola lomenej čiary. Lomená čiara je teda zložená z úsečiek $(x_i, y_i) - (x_{i+1}, y_{i+1})$ pre $i \in \{1..n-1\}$.

Môžete predpokladať, že lomená čiara sama seba nepretína a tiež žiadne tri vrcholy lomenej čiary neležia na jednej priamke.

Formát výstupu

Vypíšte "Possible" pokiaľ lomená čiara môže prejsť na druhú stranu steny (ktorou je priamka $y = 0$), cez ľubovoľne malú štrbinu v bode $(0, 0)$. V opačnom prípade vypíšte "Impossible".

Príklady

vstup

```
4
0 1
1 1
1 2
2 2
```

výstup

```
Possible
```

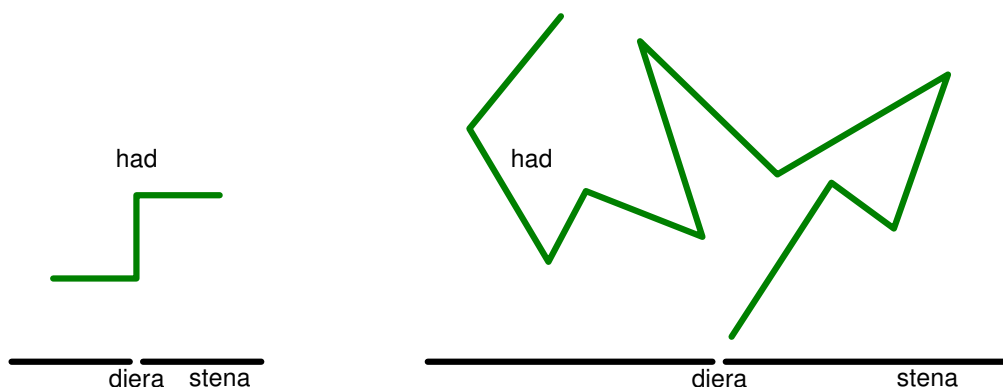
vstup

11
63 106
87 143
102 132
115 169
74 145
41 177
56 130
28 141
19 124
0 156
22 183

výstup

Impossible

Na obrázkoch môžeme vidieť nákresy ukážkových vstupov.



4. Tvorba prútikov

kat. T; 0 b za popis, 20 b za program

Alchymisti už dávno vedia ako premeniť hlinu na zlato. Nerobia to, pretože by tým mohli vážne ohroziť svetové zásoby hlíny (a v zlate sa zemiaky pestujú dosť ťažko). Preto legendárny alchymista Tomimos z Panopoly (skrátene Tomi) sústredil svoje úsilie na výrobu alchymistického prútika. Keď na stôl vedľa seba poukladáme zrnká zlata a zrnká hlíny a mávneme prútikom, každé zrno zlata sa premení na dve zrnká zlata a každé zrno hlíny sa premení na jedno zrno zlata a jedno zrno hlíny. Po nejakom počte mávnutí sa prútik pokazí a bude treba zohnať ďalší.

Navyše, keď začneme s nejakou konkrétnou postupnosťou zrníek S a opakovaným mávaním prútika vyrobíme postupnosť zrníek, ktorej podúsek je T , tak sa ten podúsek premení na nový prútik.

Tomiho teraz trápia otázky, či sa tento proces oplatí, po koľkých mávnutiach prútika sa mu podarí vytvoriť nový prútik, koľko zlata pri tom vyrobí, koľko hlíny pri tom stratí a či postupom času mu bude počet prútikov narastať alebo klesať a hlavne, či je celý proces výhodnejší ako premena pomocou kameňa mudrcov.

Pre začiatok zistíte, po koľkých mávnutiach vznikne z danej postupnosti zrníek nový alchymistický prútik.

Úloha

Zrno zlata si označíme a (aurum) a zrno hlíny b (bahno). Potom postupnosť zrníek zodpovedá reťazcu znakov a, b . Mávnutiu prútika bude zodpovedať funkcia p taká, že $p(S)$ je reťazec písmen, ktorý vznikne z S nahradením a za aa a b za ab . Napríklad $p(\text{„abaab“}) = \text{„aaabaaaaab“}$.

$p^k(S)$ označuje k -násobné aplikovanie funkcie p na S . Presnejšie $p^0(S) = S$ a $p^k(S) = p^{k-1}(p(S))$. Napríklad $p^3(\text{„b“}) = \text{„aaaaaab“}$.

Pre dané S a T nájdite najmenšie nezáporné k také, že T je podreťazec $p^k(S)$.

Formát vstupu

Vstup má dva riadky, na prvom je S a na druhom T . Každý reťazec má dĺžku aspoň 1 a najviac 200 000. Oba obsahujú len písmená a a b .

Formát výstupu

Vypíšte jedno celé číslo k , udávajúce najmenší možný vyhovujúci počet mávnutí, alebo vypíšte -1 , ak žiadne vyhovujúce k neexistuje.

Príklady

vstup	výstup
<div style="border: 1px solid black; padding: 2px;">b ab</div>	<div style="border: 1px solid black; padding: 2px;">1</div>
vstup	výstup
<div style="border: 1px solid black; padding: 2px;">ababa bab</div>	<div style="border: 1px solid black; padding: 2px;">0</div>
vstup	výstup
<div style="border: 1px solid black; padding: 2px;">a b</div>	<div style="border: 1px solid black; padding: 2px;">-1</div>

5. To kto ešte stále píše perom?

kat. T; 0 b za popis, 20 b za program

Správna odpoveď na otázku z nadpisu: všetci okrem nás programátorov. To sa nedávno vypomstilo mladému programátorovi Jarkovi. Jedného dňa spoznal krásnu mladú slečnu, s ktorou si padli do oka. Keď sa večer lúčili, mal práve Jarko vybitý mobil, a tak mu slečna perom na ruku napísala svoje telefónne číslo. Keď však Jarko prišiel domov, s hrôzou zistil, že niektoré cifry nevie prečítať.

Úloha

V tejto úlohe sa budeme hrať s bitmapami v odtieňoch šedej. Bitmapy budú mať rozmer 28×28 a každý ich pixel bude mať hodnotu od 0 (čierna) po 255 (biela). Na každej bitmape bude rukou napísaná jedna z cifier od 0 po 9, a to čiernou farbou na bielom pozadí. Príklad niekoľkých takýchto bitmáp:

8 5 9 3 1 3 3 9 1 3 5 8 6 7 9 3 1 9 2 3 7 5 0 1 1 4 7 5 9 2
8 1 0 8 1 8 9 1 9 3 0 5 2 1 7 7 7 4 6 2 7 7 5 5 4 4 7 4 0 8
0 2 0 3 2 2 0 8 0 2 4 3 0 4 9 5 3 5 2 0 1 3 2 5 1 7 9 6 8 8
0 6 7 4 3 0 6 0 9 9 1 7 8 4 6 3 2 5 4 7 7 4 1 9 6 4 7 1 3 1
0 8 5 4 8 2 0 6 5 9 6 7 8 7 6 4 4 9 3 5 9 1 4 5 1 5 9 8 3 8
8 6 0 6 6 6 5 0 1 5 5 7 0 5 4 2 5 4 5 0 0 9 5 9 3 6 1 4 6 5
5 0 5 9 1 0 3 1 8 9 9 7 7 5 0 7 1 5 8 4 8 5 8 6 0 3 8 6 6 9
9 8 4 0 5 9 4 7 7 6 7 6 8 5 4 2 6 2 2 3 3 0 4 1 6 6 2 1 1 6
2 1 5 1 7 1 6 3 2 6 4 3 5 2 4 6 0 2 3 0 4 3 4 6 2 2 6 3 9 7
1 3 2 2 3 5 1 2 3 6 3 8 6 6 7 9 3 2 4 3 9 8 7 0 4 8 6 0 1 2

Môžeš si od nás stiahnuť [zip archív](#) (13 MB) obsahujúci nasledovné dobroty:

- vstupný súbor s 50 000 bitmapami cifier
- obrázok s prvými 10 000 ciframi z toho vstupného súboru
- výstupný súbor v ktorom je ku každej bitmape zo vstupu správna cifra

Tvojou úlohou je napísať a odovzdať program, ktorý zo vstupu načíta súbor s bitmapami cifier (v rovnakom formáte ako ukázkový, vrátane prázdnych riadkov medzi bitmapami) a na výstup vypíše postupnosť cifier na bitmapách (tiež v rovnakom formáte ako ukázkový).

Hodnotenie

Tvoje riešenie bude otestované na súbore s 10 000 bitmapami cifier. (Tieto bitmapy nie sú medzi 50 000 bitmapami z ukázkového vstupu. Všetkých 60 000 bitmáp naozaj pochádza od ľudí.)

Nech x je počet bitmáp, pre ktoré dá tvoj program správnu odpoveď. Počet bodov, ktoré dostaneš, bude medián spomedzi 0, 20 a $(x - 1000)/420$, zaokrúhlený na dve desatinné miesta. (Inými slovami, ak $x \leq 1000$, dostaneš nulu, ak $x \geq 9400$, dostaneš 20, a medzi týmito dvomi hranicami počet bodov rastie lineárne.)

Upozornujeme, že tvoj program musí skončiť v časovom limite (30 sekúnd), inak nedostane body žiadne.