Zadanie celoštátneho kola súťaže ZENIT v programovaní

Kategória A a B, 12.2.2020

V prípade nejasností konzultujte záložku **Pomoc** na stránke `zenit.ksp.sk`, alebo sa spýtajte organizátorov. Úlohy sú hodnotené úplne nezávisle a samostatne, takže ich môžete riešiť v ľubovoľnom poradí. Časový limit označuje, koľko času dostane váš program pri testovaní na našom serveri (nie na vašom lokálnom počítači). Počas súťaže môžete nájsť zadania aj na webstránke. Ak by sa papierové a tlačené zadania v nejakom detaile (napríklad časovom limite) nezhodovali, tak pravdu majú zadania na **webstránke**.

A: Aké grafy?

10 bodov; časový limit: 2000 ms

V tejto úlohe si najprv prejdeme zopár definícií.

Graf G je dvojica množín (V, E), kde V je množina vrcholov (čísel) a E je množina dvojíc vrcholov, teda $E \subseteq [V]^2$ – hrana $\{a, b\}$ zaznačuje, že vrcholy a a b sú **susedné**.

Hranu spájajúcu vrchol sám so sebou nazývame **slučka**.

Graf nazývame **planárny**, ak sa dá nakresliť v rovine tak, že hrany majú prienik iba na ich koncoch. Jednoduchšie povedané, vieme na papier nakresliť vrcholy ako krúžky a hrany ako lomené čiary medzi nimi tak, že sa žiadna hrana nepretína s inou hranou.

Vrcholové farbenie grafu $G = (V, E)$ je zobrazenie $c : V \to S$, prvkom množiny S hovoríme aj **farby**.

Vrcholové k-farbenie grafu $G = (V, E)$ je také horeuvedené zobrazenie, kde $S = \{1, 2, \ldots, k\}$.

Farbenie je **regulárne**, ak všetky susedné vrcholy majú rôzne farby.

Chromatické číslo $\chi(G)$ je najmenšie k také, že G má regulárne vrcholové k-farbenie.

Nakoniec si dokážeme jednu zaujímavú vetu.

Veta (Heawood): Pre každý planárny graf G (bez slučiek) platí $\chi(G) \leq 5$.

Dôkaz (obsahuje zopár výrazov, ktoré sme si nedefinovali. Ak sa vám niektorý výraz zdá dôležitý, použite internet):

Matematickou indukciou na počet vrcholov. Každý jednoduchý planárny graf má vrchol stupňa nanajvýš 5, povedzme v. Zafarbime graf $G - v$. Tento graf má menej vrcholov ako G a teda z indukčného predpokladu platí $\chi(G) \leq 5$. Ak existuje farba nepoužitá na suseda v v G, dofarbíme v. Inak môžeme predpokladať, že susedia v v G sú $v_1, v_2, ..., v_5$ v cyklickom poradí podľa vnorenia a v_i je zafarbený farbou i. Graf indukovaný ľubovoľnými dvoma farbami je bipartitný. Ak komponent grafu $G - v$, indukovaný farbami 1 a 3 obsahujúci v_1 neobsahuje v_3, vymeníme farby v tomto komponente a vrchol v zafarbíme farbou 1. Vypíšte, či má graf párny počet hrán. Predpokladajme teda, že komponent grafu $G - v$, indukovaný farbami 1 a 3 obsahujúci v_1 obsahuje v_3. Potom existuje cesta medzi v_1 a v_3 taká, že jej vrcholy majú striedavo farby 1 a 3. Urobme rovnakú úvahu aj pre v_2 a v_4. Rovnako, komponent indukovaný farbami 2 a 4 obsahujúci v_2 obsahuje aj v_4 a existuje cesta medzi nimi na ktorej sa striedajú farby 2 a 4. Z rozmiestenia vrcholov v_1, v_2, v_3, v_4 a v potom vidíme, že existujú kružnice tvorené cestami medzi v_1-v-v_3 a v_2-v-v_4 ktoré sa križujú, čo je spor s planaritou grafu G. Teda komponent indukovaný farbami 2 a 4 obsahujúci v_2 neobsahuje v_4. Prefarbíme tento komponent a zafarbíme v farbou 2.

Táto veta od pána Heawooda je super, avšak nám nedáva návod ako zistiť, či má daný graf $\chi(G) = 5$. To je dosť škoda.

Vstup a výstup

V prvom riadku vstupu je číslo $n \leq 50$ a $m \leq 200$ – počet vrcholov a hrán grafu.

Nasledujúcich m riadkov obsahuje dve čísla $1 \leq a_i, b_i \leq n$ - hrany grafu.

Je zaručené, že uvedený graf je jednoduchý, planárny, a bez slučiek.

Vypíšte `Ano` alebo `Nie` tak, ako bolo popísané v zadaní.

Príklady

vstup

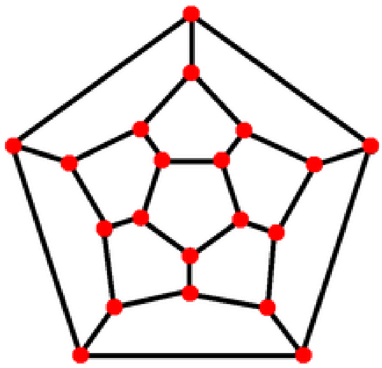
```
3 3
1 2
2 3
3 1
```

výstup

```
Nie
```

vstup	výstup

<pre>20 30
1 2
2 3
3 4
4 5
5 1
1 6
2 8
3 10
4 12
5 14
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 15
15 6
7 17
9 18
11 19
13 20
15 16
16 17
17 18
18 19
19 20
20 16</pre>
	<pre>Ano</pre>



B: Bodové odmeny

15 bodov; časový limit: 500 ms

Zenit má málo financií. A preto budú od budúceho roka súťažiaci odmeňovaní iba niekoľkými štvorčekmi čokolády. Bude to jednoduché. Ak súžažiaci a dostane menej bodov, ako súťažiaci b, tak a musí dostať menej štvorčekov čokolády, ako b. Zároveň ale chceme súťažiacich motivovať k riešeniu, kvôli čomu musí každý dostať aspoň 1 štvorček čokolády.

Zenit má málo organizátorov. A preto musíte niečo spraviť vy. Konkrétne, napíšete program, ktorý pre dané bodové zisky súžažiacich vypočíta, koľko najmenej štvorčekov čokolády budeme potrebovať na ich odmenenie.

Vstup a výstup

V prvom riadku vstupu je číslo n, počet súťažiacich. Platí, že $1 \leq n \leq 200\,000$. V prvej z troch sád zároveň platí, že $n \leq 10$. V druhom riadku vstupu je n medzerou oddelených celých čísiel a_1 až a_n, bodové zisky jednotlivých súťažiacich. Platí, že $0 \leq a_i \leq 2 \cdot 10^9$.

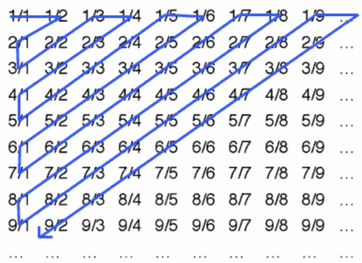
Na výstup vypíšte jedno číslo - koľko najmenej štvorčekov čokolády potrebujeme na odmenenie súťažiacich.

Príklad

vstup	výstup
6 6 2 4 1 9 7	21

C: Canton

20 bodov; časový limit: 300 ms



Zoraďme si všetky kladné zlomky do mriežky ako na obrázku, a prechádzajme nimi – začneme zlomkom $\frac{1}{1}$, druhý v poradí je $\frac{1}{2}$, tretí $\frac{2}{1}$, štvrtý $\frac{3}{1}$, a tak ďalej.

Ktorý zlomok bude v poradí n-tý?

Vstup a výstup

V prvom riadku vstupu je číslo t - počet otázok.

Každý z nasledujúcich riadkov obsahuje jedno kladné celé číslo n.

Vypíšte n-tý zlomok ak ich číslujeme vo vyššie popísanom poradí.

$1 \leq t \leq 5$

$1 \leq n \leq 10^9$

V sadách postupne rastie n.

Príklad

vstup	výstup
3 1 4 47	1/1 3/1 2/9

D: Dobré substringy

20 bodov; časový limit: 300 ms

Substring je dobrý, ak obsahuje aspoň jednu 1. Koľko je v stringu s dobrých substringov?

Vstup a výstup

Na jedinom riadku vstupu dostanete reťazec s postavený zo znakov 0 a 1.

Na jediný riadok výstupu vypíšte, koľko súvislých podúsekov reťazca s v sebe obsahuje aspoň jeden znak 1.

Hodnotenie

Platí, že $1 \leq n \leq 200\,000$. Sú 4 sady vstupov. V prvej navyše platí, že $n \leq 300$ a v druhej platí, že $n \leq 5\,000$.

Príklad

vstup	výstup
0101	8

E: Excelentný Excel

30 bodov; časový limit: 2000 ms

Šéf Korporácie Siahodlhých Písomností sa pozerá na siahodlhý hárok v Exceli. Chce v ňom nájsť veľa dôležitých údajov, ale hárok je veľmi dlhý a skrolovanie v ňom zaberá veľmi veľa času. Šéf presne vie, na ktoré riadky sa chce pozrieť a dokonca aj to, v akom poradí. Len keby to netrvalo tak dlho.

Šéfov hárok má n riadkov očislovaných od 1 po n. Na obrazovku jeho počítača sa vojde vždy k po sebe idúcich riadkov. Na začiatku sú na obrazovke zobrazené riadky 1 až k. Šéf sa chce pozrieť na m riadkov v danom poradí, tieto riadky nemusia byť rôzne.

Ak sú na obrazovke riadky i až $i+k-1$ a chceme si zobraziť riadky j až $j+k-1$, kde $i \neq j$, potrebujeme použiť jedno skrolovanie dĺžky $|i-j|$.

Nájdite najkratšiu postupnosť skrolovaní, ktorú šéf KSP musí použiť, aby si prečítal všetky žiadané riadky. Najkratšia postupnosť skrolovaní je taká, ktorá obsahuje najmenej skrolovaní a spomedzi všetkých takých je súčet dĺžok jednotlivých skrolovaní najmenší možný.

Vstup a výstup

Na prvom riadku dostanete tri celé čísla n, k, m – počet všetkých riadkov v hárku, počet riadkov, ktoré sa dajú vidieť naraz a počet riadkov, na ktoré sa chce šéf pozrieť. Platí $1 \leq k \leq n \leq 10^9$ a $1 \leq m \leq 500000$.

Nasleduje m riadkov, i-ty z nich obsahuje číslo $1 \leq r_i \leq n$ – číslo riadku, ktorý chce šéf vidieť ako i-ty v poradí.

Vypíšte dve čísla oddelené medzerou – minimálny počet potrebných skrolovaní a najmenšiu možnú preskrolovanú vzdialenosť (súčet dĺžok jednotlivých skrolovaní). Výstup nezabudnite ukončiť znakom nového riadka.

Príklady

vstup

```
20 10 2
10
20
```

výstup

```
1 10
```

Najskôr si šéf prečíta riadok 10, potom preskroluje na riadky 11 až 20 a prečíta si tie.

vstup

```
20 10 2
10
7
```

výstup

```
0 0
```

Žiadne skrolovanie nie je potrebné.

vstup

```
20 8 3
1
11
4
```

výstup

```
1 3
```

Po prečítaní prvého riadku stačí preskrolovať na riadky 4 až 11.

F: Fezjo sa hrá s dominami

30 bodov; časový limit: 300 ms

Fezjo je flákač. Furt fabrikuje fajnové fikcie, prečo nemusí programovať.
Fiškus fyzicky formuje farebnú reťaz z domín. Frfle, keď táto reťaz nie je fakt fest dlhá.
Si frajer, chceš na Fezja flexiť natoľko, že bude ten fidlikár fňukať?
Tak poflirtuj s nápadom, že postavíš dlhšiu reťaz ako on.

Úloha

Dominá pozostávajú zo štvorčekov dvoch farieb - bielej (B) a čiernej (C). Dve dominá vieme položiť vedľa seba v reťazi, ak sú ich dotýkajúce štvorčeky rovnakej farby.

Teda z domín BBCB, CC a BCB vieme postaviť reťaz BCB|BBCB, dĺžky 7 štvorčekov.

Dominá **nesmieme** otáčať pri ukladaní do reťaze. Domino BC teda nemôžeme otočiť a použiť ako CB.

Na vstupe dostanete popis domín, ktoré má Fezjo k dispozícií. Zistite, akú dlhú reťaz dokáže postaviť.

Vstup a výstup

V prvom riadku vstupu je číslo n – počet domín.

Každy z nasledujúcich n riadkov obsahuje popis jedného domina – reťazec z písmen C a B.
Vypíšte dĺžku (počet znakov) najdlhšej reťaze, ktorá sa z nich dá vytvoriť.
Dominá sú dlhé aspoň 1 a najviac 100 znakov.
n je postupne 10, 20, 1 000.

Príklady

vstup

```
3
BBCB
CC
BCB
```

výstup

```
7
```

vstup

```
4
CB
BBCC
BCC
BCBBC
```

výstup

```
11
```

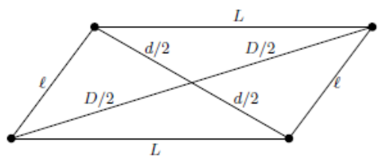
G: Gréci vo hviezdach

40 bodov; časový limit: 3000 ms

Možno ste niekedy rozmýšľali nad tým, aké látky to tí starovekí Gréci mohli užívať, ktoré mali také silné fantasmagorické účinky, že vo štyroch hviezdach umiestnených v akom-takom obdĺžniku videli škorpióny a levy a bohviečo.

Ale možno, že to nie je až tak pritiahnuté za uši. Čo keď rôzne rozostavenia hviezd sú veľmi vzácne? Napríklad taký rovnobežník?

Rovnobežník je taký štvoruholník, ktorého protistojace strany majú rovnaké dĺžky. Ekvivalentne ho vieme definovať ako konvexný štvoruholník ktorého diagonály sa pretínajú v svojich stredoch. Doleuvedený obrázok ilustruje obe definície, pre rovnobežník so stranami dĺžky L a l, a diagonálami dĺžky D a d.



Úloha

Hviezdne nebo si môžeme reprezentovať ako rovinu, a hviezdy ako body v nej. Vašou úlohou je spočítať, koľko rôznych rovnobežníkov sa na nebi nachádza. Dva rovnobežníky považujeme za rôzne, ak nezdieľajú aspoň jednu hviezdu.

Vstup a výstup

V prvom riadku vstupu je číslo n – počet hviezd.

V nasledovných n riadkoch sú dve celé čísla x_i, y_i – súradnice i-tej hviezdy. Žiadne dve hviezdy nie su na rovnakých súradniciach.

Platí $-10^8 \leq x_i, y_i \leq 10^8$.

n je aspoň 4 a postupne maximálne $47, 300, 1000$.

Príklady

vstup

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

výstup

```
3
```

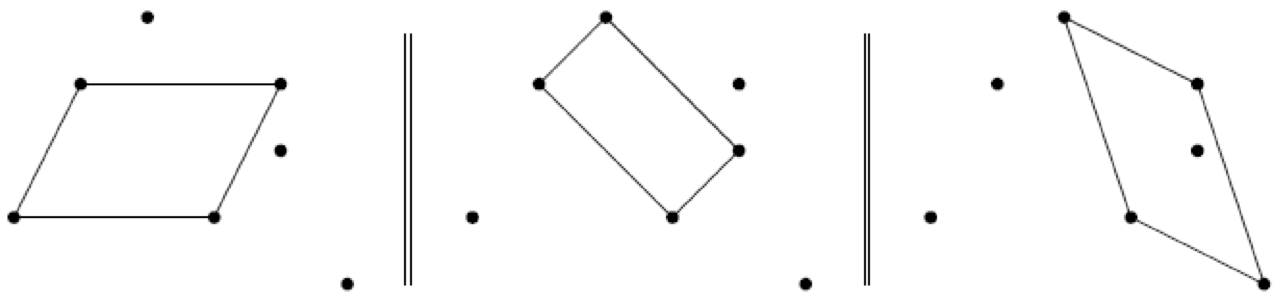
vstup

```
11
0 -1
1 1
1 -1
0 2
0 0
1 0
-1 1
-1 -1
0 1
1 2
-1 0
```

výstup

```
44
```

Prvý príklad a rovnobežníky v ňom.



H: Heavy-Light Decomposition

40 bodov; časový limit: 500 ms

Keď ste upratovali pivnicu, našli ste pod hromadou prachu a zaváraninových pohárov aj nádhernú fotografiu holokarstu a vcelku zachovaný binárny strom. Stromček sa vám naozaj páči, každý vrchol má na sebe totiž napísané dve čísla a celý strom je generovaný iba dvoma jednoduchými pravidlami - ak má vrchol na sebe napísané čísla A a B tak:

1. ak je A=B tak je vrchol list
2. inak má jeho ľavý syn napísané čísla A a $(A+B)/2$ a jeho pravý syn $(A+B)/2+1$ a B

List nazveme osamelým, ak nie sú obaja synovia jeho otca listy. Viete čo máte robiť – spočítajte ich.

Vstup a výstup

Na prvom riadku je jedno celé číslo T - počet otázok.

Nasleduje T riadkov. Na i-tom riadku sú dve celé čísla A_i a B_i, udavajúce dve čísla napísané v koreni stromu pre i-tu otázku.

Na výstup vypíšte T riadkov, pričom na i-tom riadku bude jedno celé číslo - odpoveď na i-tu otázku.

Obmedzenia

Vo všetkých testovacích sadách bude platiť $0 \leq T \leq 10^5$ a $0 \leq N \leq 10^{18}$, pričom $0 \leq A_i \leq B_i \leq N$ pre všetky i. Navyše, obmedzenia na N a T budú v jednotlivých sadách nasledovné:

```
|   \logT|2|3|4|5|
|logN\   +-+-+-+-|
|    2   |X| | |X|
|    4   | |X| |X|
|    7   | | |X|X|
|   18   |X|X|X|X|
```

kde X reprezentuje existenciu jednej sady s danými obmedzeniami (teda napríklad posledný riadkok a stĺpec znamená sadu s $T = 10^5, N = 10^{18}$). Bude teda 10 sád, za každú sadu získate 10% celkového počtu bodov.

Upozorňujeme, že vstupné súbory môžu byť vcelku masívne a preto odporúčame použiť rýchle načítavanie a vypisovanie vo vami zvolenom jazyku.

Príklady

vstup

```
2
2 4
4 5
```

výstup

```
1
0
```

(2,4) -> (2,3),(4,4); (2,3) -> (2,2), (3,3). (4,4) je jediný osamelý list.

I: Ideme cestovať

55 bodov; časový limit: 1000 ms

Kubík je veľký pocestný obchodník s elektronikou. Jeho povolanie od neho vyžaduje nadšenie, veľkú flexibilitu a obetavosť. Nadovšetko od neho však vyžaduje schopnosť prepraviť sa medzi ľubovoľnými dvoma mestami v priebehu jediného dňa a po poslednom fiasku s meškajúcim vlakom sa Kubík napokon rozhodol zaobstarať si novú Teslu. Môže si to predsa dovoliť! Nechce však ani plytvať peniazami a zbytočne si kúpiť priveľkú batériu. Počas jedného dňa je ochotný sa najviac C krát zastaviť na nabíjacej stanici, pričom sa vždy zastaví nabiť pred tým, ako v ten deň vyrazí na cestu medzi mestami. Mestá tvoria súvislý graf s N vrcholmi a M ohodnotenými hranami a v každom meste je nabíjacia stanica, na ktorej si môže batériu nabiť na ľubovoľnú úroveň.

Aký najmenší dojazd môže mať batéria, aby sa Kubík vedel dostať medzi ľubovoľnými dvoma mestami na najviac C nabití?

Vstup a výstup

Na prvom riadku vstupu je jedno celé číslo T - počet otázok.

Nasleduje T otázok. Pre každú otázku: Sú na prvom riadku 3 celé čísla N, C a M. Nasleduje M riadkov, na i-tom riadku sú 3 celé čísla a_i, b_i a d_i, ktoré reprezentujú cestu dĺžky d_i medzi mestami a_i a b_i. Každá neusporiadaná dvojica (a_i, b_i) sa vyskytne najviac raz, pričom pre každé i je a_i rôzne od b_i.

Vo všetkých testovacích sadách platí $0 \leq T \leq 10$, $0 \leq N \leq 100$, $0 \leq C \leq 1000$, pričom $0 \leq a_i, b_i \leq N - 1$ a $0 \leq d_i \leq 10^9$ pre všetky i. Navyše, v prvých 2 sadách budú $0 \leq d_i \leq 100$ a N bude v jednotlivých sadách postupne najviac $5, 10, 30, 50, 100$. Celkovo bude teda 5 sád, každá za 20% celkového počtou bodov.

Na výstup vypíšte T riadkov, na každom riadku najmenšiu kapacitu batérie pre prislúchajúcu otázku.

Príklady

vstup	výstup

<pre>2
4 2 4
0 1 100
3 0 400
1 2 200
2 3 300
10 2 15
3 8 355
4 9 113
5 7 235
7 9 979
8 5 462
0 5 411
0 1 113
1 2 314
9 6 402
6 8 431
2 3 271
3 4 141
4 0 173
1 6 855
2 7 921</pre>
	<pre>300
	688</pre>

J: Juchú Vianoce

60 bodov; časový limit: 100 ms

Blížia sa vianoce a preto treba ozdobiť stromček. Vianočný stromček je Trie - vyhľadávací strom v ktorom hrany reprezentujú znaky a teda každý vrchol reprezentuje prefix reťazca daného cestou k nemu, pričom každý vrchol reprezentuje unikátny prefix (cs.wikipedia.org/wiki/Trie).

Ozdobovanie stromčeka je v tvojej rodine dlhoročnou tradíciou do ktorej sa všetci zapájajú. Ozdobovací rituál prebieha nasledovne: každý člen rodiny si do ruky zoberie svetelnú reťaz a všetci naraz ich šmaria na stromček. Samozrejme, svetelná reťaz je perfektne náhodný binárny reťazec dĺžky L (reťazce sa medzi hodmi môžu opakovať) a keď pristane na strome, znamená to, že sme ju vložili do Trie.

Aký je očakávaný počet vrcholov vianočného stromčeka, ak má tvoja rodina N členov?

Vstup a výstup

Na prvom riadku je jedno celé číslo T - počet otázok.

Nasleduje T riadkov. Na i-tom riadku sú dve celé čísla n_i a l_i, udávajúce i-tu otázku tvaru "Aký je očakávaný počet vrcholov Trie ak do nej vložím n_i náhodných binárnych reťazcov dĺžky l_i?".

Na výstup vypíšte T riadkov, pričom na i-tom riadku bude jedno číslo - odpoveď na i-tu otázku. Odpovede vypisujte zaokrúhlené na 2 desatinné miesta, vždy vypíšte práve 2 (slovom dve) desatinné miesta.

Obmedzenia

Vo všetkých testovacích sadách bude platiť $0 \leq T \leq 50$, $1 \leq N, L \leq 300$ a $1 \leq n_i \leq N$ a $1 \leq l_i \leq L$ pre všetky i. Navyše v jednotlivých testovacích sadách bude obmedzenie na N a L postupne 5, 10, 20, 100 a 300 - za každú sadu dostanete 20% bodov.

Príklady

vstup	výstup

<pre>2
1 3
2 2</pre>
	<pre>4.00
	4.25</pre>

V prvej otázke existuje 8 možných reťazí, po vložení ľubovoľnej z nich bude mať strom 4 vrcholy.

K: Keltský nápis

65 bodov; časový limit: 3000 ms

Kráľovská Skupina Prieskumníkov (KSP) pri nedávnych vykopávkach odhalila staroveký nápis vytesaný do kameňa, ktorý na mieste zanechala keltská civilizácia pred viac ako tisíc rokmi. Súčasťou tohto nápisu bol aj vzorec, ktorého výsledok bol rovný posvätnému číslu. Zo vzorca sa však zachovali už iba čísla, žiadne operátory už po mnohých storočiach nie sú čitateľné.

KSP sa rozhodla, že sa pokúsi nápis zrekonštruovať, avšak nie je to vôbec jednoduché. Z iných zdrojov sa im podarilo zistiť posvätné číslo. Pri výskumoch sa zistilo, že starovekí Kelti používali tieto 4 operátory:

- $+$ – znamená sčítanie presne tak, ako u nás
- $-$ – znamená odčítanie presne tak, ako u nás
- . – hodnotou výrazu $a.b$ je $a + 2b$
- $\sim$ – hodnotou výrazu $a \sim b$ je $a - 2b$

Zátvorky ešte Kelti nepoznali, všetky operátory majú rovnakú prioritu a výraz sa vyhodnocuje zľava doprava. Vašou úlohou je pomôcť prieskumníkom zrekonštruovať staroveký nápis. Váš program dostane na vstupe čísla vo vzorci v poradí, v akom sú vytesané, a posvätné číslo Keltov. Medzi každé dve susedné čísla doplňte práve jeden z uvedených operátorov tak, aby výsledok výrazu bol rovný posvätnému číslu.

Vstup a výstup

Na prvom riadku sú dve celé čísla n (počet čísel vo vzorci) a t (posvätné číslo Keltov), pričom platí $2 \leq n \leq 22$ a $0 \leq |s| \leq 10^{16}$.

Na druhom riadku je n kladných celých čísel neprevyšujúcich 10^{15}, oddelených medzerou – jednotlivé čísla v objavenom vzorci.

Na výstup vypíšte postupnosť $n - 1$ keltských operátorov, ktoré keď v poradí, v akom vypíšete, vložíme medzi zadané čísla, výsledkom bude číslo t. Medzi operátory nevypisujte žiadne medzery. Ak existuje viac riešení, vypíšte ľubovoľné. Ak neexistuje žiadne, vypíšte namiesto toho reťazec `neexistuje`.

Príklady

vstup

```
5 3
1 4 2 5 3
```

výstup

```
.-~.
```

Platí $1 + 2 \cdot 4 - 2 - 2 \cdot 5 + 2 \cdot 3 = 1 + 8 - 2 - 10 + 6 = 3$. *Iným správnym riešením by bolo aj* $. \sim -+$, *pretože* $1 + 2 \cdot 4 - 2 \cdot 2 - 5 + 3 = 3$.

vstup

```
2 13
10 7
```

výstup

```
neexistuje
```

L: Ľahké to nebolo

70 bodov; časový limit: 500 ms

Zadanie tejto úlohy je totožné so zadaním úlohy Izbové monštrum z krajského kola 2018

Dávid sa z nudy začal prehrabávať v hŕbe starých súčiastok, televízorov, kalkulačiek, procesorov, … a čo nevidieť, mal zostrojený hračkársky počítač. Nazval ho "Izbové monštrum".

Dávidov počítač funguje nasledovne. Pracuje primárne s dvojicami čísel, a vie s nimi robiť niekoľko rôznych operácii, ktoré popíšeme neskôr. Pamäť počítača je indexovaná prirodzenými číslami, pričom na začiatku je na indexe 0 hodnota (x, y) a ostatné pamäťové miesta sú voľné. Je chyba, keď pri výpočte používame hodnotu na voľnom pamäťovom mieste.

Podporované operácie sú tri. Každá vezme hodnoty z niekoľkých pamäťových miest, prípadné číselné argumenty, spracuje ich, a výsledok uloží na prvé voľné pamäťové miesto. Operácie sú:

- Zoberie hodnotu (a, b), kladné číslo l a vráti $(a + l, b + l)$.
- Zoberie hodnotu (a, b), kde a aj b sú párne. Vráti $(\frac{a}{2}, \frac{b}{2})$.
- Zoberie dve (nie nutne rôzne) hodnoty (a, b) a (c, d), ktoré spĺňajú $b = c$. Vráti (a, d).

Dávida teraz zaujíma, či vie nejakou postupnosťou operácii získať hodnotu (x', y'). Hľadá teda takú postupnosť operácii, po ktorej bude v poslednom použitom pamäťovom mieste hodnota (x', y').

Vstup a výstup

Vstup bude pozostávať s viacerých testovacích prípadov. Na prvom riadku je číslo t, určujúce ich počet. Nasledujú popisy jednotlivých prípadov, každý z nich na samostatnom riadku. Vo všetkých vstupoch okrem vzorového platí $t = 100$.

Popis prípadu obsahuje štyri kladné celé čísla oddelené medzerou: x, y, x' a y'. Prvé dve určujú hodnotu na pamäťovom mieste 0, s ktorou Dávid začína. Ďalšie dve určujú cieľovú hodnotu, ktorú chce Dávid dosiahnuť. Tieto štyri čísla neprevýšia 10^9.

Ak sa cieľová hodnota nedá dosiahnuť, vypíšte riadok obsahujúci číslo -1. Ak sa dá dosiahnuť, vypíšte v prvom riadku počet operácii, ktoré použijete. Potom vypíšte popisy jednotlivých operácii, každý na samostatnom riadku. Popis každej operácie začína typom operácie (1, 2 alebo 3), nasledujú čísla pamäťových miest, ktoré sú argumentami pre tú operáciu (môžu byť aj rovnaké), a potom prípadné číselné argumenty. Všetky čísla v popise operácie sú oddelené medzerami.

Medzi každými dvoma odpoveďami (na jednotlivé testovacie prípady) nech je prázdny riadok.

Ak existuje viacero správnych postupností operácii, smiete vypísať ľubovoľnú z nich. Nemusí byť najkratšia, musíte ju ale stihnúť vypísať v časovom limite.

Príklad

vstup

```
2
1 1 2 3
3 5 100 101
```

výstup

```
-1

3
1 0 1
2 1
1 2 98
```

V druhom prípade by sme vedeli dosiahnuť cieľové hodnoty aj na 2 operácie.