



Úlohy 2. kola zimnej časti

Termín odoslania riešení tohto kola je pondelok 9. januára 2016.

1. Zuzkino sudoku

kat. Z; 6 b za popis, 4 b za program

Zuzka rada rieši a ešte radšej vytvára sudoku. Vo svojom rodnom meste dokonca vytvárala každý týždeň jedno sudoku pre miestny časopis.

Jedného dňa jej napísal vydavateľ celoslovenského denníka s ponukou, či by nechcela sudokami zásobovať celé Slovensko každý deň. Na každý deň by musela vytvoriť ľahkú, strednú aj ťažkú variantu. To je ale veľa práce, však? Ak by Zuzka ponuku prijala, síce by si zarobila, no nemala by čas na školu, spánok a kamarátov¹.

Zuzka je ale programátorka, a preto nebude sudoku vyrábať ručne, ale programom. Nie je to však také jednoduché. Pri tvorbe si totiž treba dať záležať na tom, aby mali riešitelia z riešenia sudoku dobrý zážitok. Sudoku, v ktorom chýbajú iba jednotky, je veľmi nezábavné. Také, do ktorého treba počas riešenia doplniť rôzne čísla je rozhodne zaujímavejšie.

Pomôžte Zuzke zistiť, ktoré sudoku sú tie zaujímavejšie. Napíšte program, ktorý pre dané sudoku zistí, koľko ktorých čísel treba do sudoku doplniť pre jeho vyriešenie.

Úloha

Sudoku sa hrá nasledovne. Dostanete tabuľku s 9×9 políčkami. Tabuľka je rozdelená na 3×3 bloky po 3×3 políčkach. Na každom políčku je buď číslo od 1 po 9 alebo je políčko voľné. Úlohou v sudoku je doplniť čísla na všetky prázdne políčka tak, aby sa v žiadnom riadku, stĺpci ani v bloku nezopakovalo žiadne z čísel 1 až 9 viac ako raz.

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

Vašou úlohou je pre daný popis sudoku zistiť, koľko ktorých čísel doň treba doplniť.

Formát vstupu

Na vstupe dostanete 9 riadkov. V každom z nich bude 9 čísel v rozsahu od 0 po 9 oddelených medzerami. Čísla 0 reprezentujú nevyplnené políčka sudoku.

Môžete predpokladať, že sudoku zo vstupu sa dá vyriešiť – teda napríklad v žiadnom riadku, stĺpci ani v bloku sa nezopakuje to isté číslo (okrem nuly).

Formát výstupu

Vypíšete 9 riadkov. Ak riadky očísľujeme od 1 po 9, na i -tom z nich sa má nachádzať jedno celé číslo – počet čísel i , ktoré treba do sudoku doplniť.

¹Veľa ľudí sa aj bez vytvárania sudoku sťažuje, že majú čas len na dve zo spomenutých troch vecí.

Príklad

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

Jednotka chýba v piatom a v šiestom riadku. Dvojky sú všade tam, kde majú byť, ...

2. Zabudnutá chata

kat. Z; 6 b za popis, 4 b za program

Kde bolo tam bolo, tam, kde sa piesok lial a nesypal, kde vtáky spievali pop-punkové balady, v krajine bohatej na prírodu sa nachádzal borovicový les. V strede tohto lesa sa nachádzala zabudnutá chata. Vedúci KSP (kúlový spolok programátorov) sa o tejto chate dozvedeli cez Google, ktorý ju označil ako lacnú. Okamžite sa v nej rozhodli spraviť si veľkolepú oslavu, na ktorej sa chce zúčastniť úplne každý.

Ako to však vo svete chodí, nič nie je zadarmo, a aj prenájom tejto chaty niečo stojí. Každý vedúci má našetrenú nejakú sumu peňazí, z ktorej je ochotný prispieť ľubovoľne veľa (ale nie viac). Problém je však v tom, že na chatu chce dať každý presne rovnakú sumu, ako na ňu dajú ostatní².

Keďže je táto chata taká zabudnutá, pani domáca nie je veľmi zvyknutá na ľudí a dokáže byť pomerne nevrlá³. Vedúci si ju nechcú pohnevať napríklad tým, že jej budú platiť v centoch, a preto radšej každý zaplatí sumu v celých eurách, aj keby mali celkovo zaplatiť viac, ako je potrebné.

Úloha

Pre danú cenu chaty c a osobné finančné limity každého vedúceho s_i zistíte maximálny počet ľudí, ktorí sa môžu zúčastniť tejto chaty. Dokopy musia byť schopní chatu zaplatiť a každý z nich musí zaplatiť rovnakú celočíselnú sumu (pričom nikto nepresiahne svoj limit). Nezaujímá nás, či za cenu pridania ďalšieho vedúceho zaplatíme dokopy o niečo viac.

Formát vstupu

Na prvom riadku dostanete postupne celé čísla n, c ($1 \leq n \leq 150\,000$, $0 \leq c \leq 10^9$) – počet vedúcich a cenu chaty. Na nasledujúcom riadku bude n medzerou oddelených čísel $0 \leq s_i \leq 10^9$. s_i reprezentuje najväčšiu sumu, ktorú je ochotný zaplatiť i -ty vedúci.

Medzivýpočty sa nemusia zmestiť do klasických celočíselných premenných. Odporúčame preto používať typy `long long` v `C++`, prípadne `int64` v `Pascale`.

Formát výstupu

Vypíšte jedno číslo – maximálny počet vedúcich, ktorí sa chaty zúčastnia. Výstup nezabudnite ukončiť znakom nového riadku.

Príklad

vstup	výstup
3 47 47 1 42	2
10 7 0 1 0 0 2 1 1 1 0 1	0

Zaplatia to prvý vedúci s tretím, každý po 24 eurách.

Hoci by vedúci vedeli dať dokopy 7 eur, každý musí zaplatiť rovnako veľa. Preto vedúci zaplatiť nevedia a na chatu nepôjde nikto.

²Bolo by asi nefér, aby Mišo cvakal všetko za Žabu.

³Podľa tej jednej recenzie, čo Google našiel.

3. Zajačí problém

kat. Z; 6 b za popis, 4 b za program

V Krajine Slepáčich Paprčiek (ďalej už len KSP) nastal čas zimy. Hlavnou náplňou práce v tejto krajine je chov sliepok a pestovanie špeciálnej mrkvy. Mrkva pestovaná v KSP sa od ostatných druhov odlišuje tým, že žiadne dve mrkvy nenarastú rovnako veľké. Keďže sliepky cez zimu nevyháňajú toľko, aby celú krajinu uživil a mrkvu už nikto nechce⁴, rozhodla sa vláda využiť svoju špeciálnu mrkvu iným spôsobom. Po dlhom rokovaní sa rozhodlo, že sa kúpia špeciálne zajace typu T2, ktoré sú známe predovšetkým vďaka svojmu vyberavému vkusu v oblasti mrkvy. Problém však nastal, keď sa rozhodli zajace nakrmiť. Niektoré zajace svoje mrkvy nejedli a iba škaredo zazerali na zajace vedľa seba.

Našťastie bola k zajacom doručená aj príručka chovateľa, z ktorej obyvatelia zistili, že kŕmenie je u zajacov T2 veľmi háklivou kultúrnou záležitosťou. V závislosti na veku, pohlaví a postavení v zajačej hierarchii sú niektoré zajace ochotné jesť iba vtedy, keď má zajac pred nimi väčšiu mrkvu. Iné zajace zasa budú jesť, iba keď má zajac pred nimi menšiu mrkvu.

Po zistení tejto informácie už dokázali obyvatelia zajace nakrmiť. Na druhý deň sa ale zajace zoradili na kŕmenie v inom poradí a obyvatelia zostali opäť zaskočení. Vláda sa teda rozhodla siahnuť po trvalom⁵ riešení a využiť zostávajúce financie rozumne. Rozhodla sa teda najat' si vás na vyriešenie tohto problému.

Úloha

Počet zajacov si označme n . Mrkvy, ktorými chcú v daný deň Slepáčopaprčkanka nakrmiť zajace, si očísľujeme od najmenej po najväčšiu číslami $1, 2, \dots, n$.

Dostanete reťazec tvorený $n - 1$ znakmi ' $<$ ' (menší) a ' $>$ ' (väčší), ktoré určujú vzťah medzi veľkosťami mrkvy susedných zajacov. Vašou úlohou je nájsť takú permutáciu čísel 1 až n , v ktorej po sebe idúce členy spĺňajú zadané nerovnosti (permutácia čísel 1 až n je taká postupnosť čísel z rozsahu 1 až n , ktorá každé z čísel od 1 po n obsahuje presne raz).

Formát vstupu

Na vstupe sa nachádza jeden riadok skladajúci sa z $n - 1$ znakov " $<$ " a " $>$ ". Platí $2 \leq n \leq 10^6$.

Formát výstupu

Vypíšte jeden riadok a v ňom n medzerami oddelených čísel: hľadanú permutáciu. Vždy existuje aspoň jedno riešenie. Ak existuje viac riešení, vypíšte ľubovoľné z nich.

Príklad

vstup	výstup
>>>>	5 4 3 2 1
<i>Platí $5 > 4, 4 > 3, 3 > 2, 2 > 1$</i>	
vstup	výstup
><<<	4 2 3 1 5

Platí $4 > 2, 2 < 3, 3 > 1, 1 < 5$. Tento vstup má rôzne možnosti riešenia – správnym riešením je napríklad aj $2\ 1\ 4\ 3\ 5$.

4. Zabudnuté algoritmy

kat. Z a O; 9 b za popis, 6 b za program

Hodobox je už vysokoškolák, a preto bol tento rok prvýkrát na školskom kole súťaže ACM. A čo sa stalo – nestalo, predbehla ho stredoškolačka a ešte aj chlapec so zápalom pľúc! Teraz sa Hodobox snaží prejsť si všetky úlohy odznova a skúsiť si ich preriešiť. Má však malý problém. Niečo sa stalo a zabudol všetky algoritmy. Nevie už nakódiť ani len binárne vyhľadávanie, ktoré sa naučil na [letnej škole](#).

Rozhodol sa teda, že sa všetky potrebné algoritmy znova naučí. Každá úloha sa dá definovať ako množina algoritmov, ktoré sú potrebné na jej vyriešenie. Pre jednoduchosť si algoritmy očísľujeme od 1 po $1\,000\,000$ (napríklad 47 je binárne vyhľadávanie).

Nie je to však až také jednoduché. Rôzne úlohy vyžadujú znalosť rôznych algoritmov do rôznej hĺbky. Napríklad taký merge-sort – v jednej úlohe ním len usporiadame prvky, v inej však spočítame počet inverzií. Úloha má teda ku každému algoritmu, ktorý je potrebný na jej vyriešenie priradenú ešte aj potrebnú úroveň znalosti tohto algoritmu.

⁴Všetci vegetáriani z Krajiny Mäsozravých Slonov sa totiž na zimu sťahujú do teplých krajov, kam sa mrkva kvôli diplomatickým problémom nevyváža.

⁵samozrejme úspornom

Hodobox má so sebou Paulínkinu príručku 'How to win IOI for dummies'. V nej si Paulínka spísala všetky algoritmy od výmyslu sveta. Hodobox sa nimi teraz prehrýza a veľmi by ho zaujímalo, kedy už bude konečne schopný naprogramovať všetky úlohy zo školského kola ACM.

Úloha

Na vstupe dostanete popis školského kola, ktoré sa skladá z n úloh. Pre každú úlohu dostanete zoznam algoritmov potrebných na jej vyriešenie (číslo algoritmu a úroveň potrebnú na vyriešenie tejto úlohy).

Potom nasleduje p Hodoboxových akcií, kde si vždy zlepši chápanie nejakého algoritmu o istú úroveň. Na začiatku Hodobox nevie nič (jeho úroveň znalosti každého algoritmu je 0).

Vašou úlohou je po každej akcii vypísať, koľko úloh zo školského kola je už Hodobox schopný vyriešiť.

Formát vstupu

Na prvom riadku vstupu je počet úloh n ($1 \leq n \leq 10^6$).

Nasleduje n popisov úloh. Popis i -tej úlohy začína číslom a_i na samostatnom riadku – počet algoritmov nutných na vyriešenie i -tej úlohy ($0 \leq a_i < 10^6$). Potom nasleduje a_i riadkov s popismi jednotlivých algoritmov potrebných pre túto úlohu. Každý z nich obsahuje dve čísla x ($1 \leq x \leq 10^6$) – číslo daného algoritmu a y ($1 \leq y \leq 10^9$) – potrebná úroveň znalosti tohto algoritmu.

Zároveň platí, že $\sum a_i \leq 10^6$.

Potom nasleduje jeden riadok s číslom p – počet akcií, kedy sa Hodobox učí nejaký algoritmus. Platí, že $p \leq 10^6$. Nasleduje p riadkov. V každom sú dve čísla c_j a u_j , kde c_j je číslo algoritmu a u_j znamená, o koľko sa Hodoboxovi zlepšila úroveň znalosti algoritmu c_j . Platí že $1 \leq c_j \leq 10^6$ a $1 \leq u_j \leq 10^9$.

Vstupy v tejto úlohe budú **veľmi veľké**. V C++ preto odporúčame použiť miesto knižnice `<iostream>` knižnicu `<cstdio>`. Ak chcete používať `<iostream>`, vložte si na začiatok programu nasledovné príkazy, ktoré urýchlia vstup a výstup: `ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);` Na vypisovanie nového riadku tiež nepoužívajte `endl`, ale znak nového riadku `'\n'`.

V pomalších jazykoch sa nemusí dať získať plný počet bodov na testovacích vstupoch. Za optimálne riešenie však stále viete získať plný počet za popis riešenia.

Formát výstupu

Na výstupe bude p riadkov. Na j -tom riadku bude jedno číslo – počet úloh ktoré vie Hodobox vyriešiť, keď sa už zlepšil v algoritmoch c_1 až c_j o príslušné úrovne.

Príklad

vstup

```
3
1
14 1
0
2
13 3
14 1
2
13 3
14 1
```

výstup

```
1
3
```

Už na začiatku vie Hodobox vyriešiť druhú úlohu, pretože na ňu netreba vedieť nič. Po prvej akcii si zlepši chápanie algoritmu 13 z 0 na 3, čo mu nepomôže k ďalšej úlohe, takže výstup je 1. Druhou akciou si zlepši chápanie algoritmu 14 z 0 na 1, čo mu pomôže vyriešiť prvú aj tretiu úlohu.

vstup

```
3
2
47 5
42 3
3
42 13
13 4
7 6
1
47 2
5
47 1
47 3
42 12
13 5
47 12
```

výstup

```
0
1
1
1
2
```

Hodobox nevie pred prvou akciou vyriešiť ani jednu úlohu. Po prvej akcii sa binárne vyhľadávanie (algoritmus 47) naučí na úroveň 1. To mu ale nepomôže k žiadnej úlohe, takže výstup je 0.

Ďalšou akciou si zlepši znalosť vyhľadávania na 4. Preto vie vyriešiť tretiu úlohu, ale nič viac. Potom sa naučí algoritmus 42 na úroveň 12, čo mu stále nepomôže k ďalším úlohám. Potom si zlepši algoritmus 13 z nuly na 5, takisto mu to nepomôže. Na prvú úlohu ma slabú znalosť algoritmu. č. 47 a na druhú zas algoritmu 7.

Poslednou akciou sa naučí algoritmus 47 na úroveň 16 (4 + 12), takže už bude vedieť vyriešiť prvú úlohu.

5. Obiehanie

kat. Z a O; 10 b za popis, 5 b za program

Pred pár týždňami sa uskutočnilo školské kolo súťaže Zenit v programovaní a mnohí z vás sa ho pravdepodobne zúčastnili. Mohli ste si na ňom všimnúť, že po odovzdaní úlohy ste museli chvíľku počkať na jej vyhodnotenie. Takú malú, 20-minútovú chvíľku. Táto chvíľka vznikla preto, lebo testovač dostal úžasný nápad, ako robiť vyhodnocovanie odovzdaných riešení. Testovať riešenia v tom poradí, v akom prišli sa mu zezdalo príliš trápne a určite neoptimálne, preto sa rozhodol usporiadať turnaje.

Ako to funguje? Najprv treba počkať, kým riešitelia odovzdajú dosť veľa riešení. Každému riešeniu je potom pridelená jeho úžasnosť⁶. Následne sa riešenia zoradia vedľa seba v takom poradí, v akom prišli a začne sa turnaj. Ide o klasického turnajového pavúka – najprv súperí prvé riešenie s druhým, potom tretie so štvrtým a tak ďalej. To úžasnejšie z dvojice riešení vyhrá a postupuje do ďalšieho kola. V druhom kole súperia už len víťazi predchádzajúceho kola rovnakým spôsobom. Takto to pokračuje, až kým nezostane len to úplne najúžasnejšie riešenie, ktoré sa naozaj aj otestuje.

Zenitu sa zúčastnil aj Romanko. Porozhliadal sa okolo seba a už na prvý pohľad vedel, ako úžasné riešenia vyprodukujú ostatní riešitelia. Dokonca aj presne vedel, kedy tieto riešenia pošlú na otestovanie a teda ako budú usporiadané počas turnaja. Romanko by teraz chcel odovzdať svoje riešenie a mať ho aj v rozumnom čase⁷ otestované. Jeho riešenie ale určite nie je dosť úžasné na to, aby mohlo vyhrať turnaj.

Romanko ale vie, že testovač je mysteriózna entita a dá sa rôzne ovplyvňovať. Ak mu človek správne obetuje horalku⁸, je možné, že jeho riešenie prejde do ďalšieho kola aj keď nebolo to úžasnejšie. Romanko sa teraz potrebuje vopchať medzi ostatné riešenia (napríklad odovzdá čo najskôr a bude teda v turnaji ako prvý) a rád by vedel, koľko horaliek bude musieť obetovať, ak chce vyhrať turnaj a začína na danej pozícii. Môžete mu s tým pomôcť?

Úloha

Na vstupe sú dané úžasnosti $n - 1$ riešení ostatných riešiteľov v takom poradí, v akom budú na testovači. Taktiež je zadaná úžasnosť Romankovho riešenia. Pre každú pozíciu, na ktorú sa môže Romanko vopchať chceme zrátať, koľko horaliek musí obetovať, aby jeho riešenie vyhralo turnaj a bolo teda otestované najskôr.

⁶zjavne padajúci bruteforce v pythone veľmi úžasný nie je, ale taká binárne vyhľadávaná, lazy-loadingová geometria už stojí za to

⁷hneď

⁸dá ju najbližšiemu vedúcemu KSP

Formát vstupu

Na prvom riadku sa nachádza číslo n – počet riešení a k – úžasnosť Romankovho riešenia. V druhom riadku sa nachádza $n - 1$ rôznych celých čísel $a_1 \dots a_{n-1}$ – úžasnosti riešení ostatných riešiteľov. Platí $1 \leq n, k, a_i \leq 2^{20} = 1\,048\,576$ a že n je mocninou 2. Taktiež platí, že úžasnosti riešení (aj Romanovho) sú navzájom rôzne.

Formát výstupu

Vypíšte n medzerou oddelených čísel, pričom i -te číslo udáva počet horaliek, ktoré Romanko potrebuje obetovať, ak by jeho riešenie začínalo na i -tej pozícii.

Príklad

vstup	výstup
<pre>4 2 1 3 4</pre>	<pre>1 1 2 2</pre>

V tomto príklade Romanko potrebuje obetovať pre prvé a druhé miesto 1 horalku. Najprv sa jeho riešenie stretne s riešením úžasnosti 1 a ľahko sa dostane do ďalšieho kola. Tam sa ale stretne s riešením úžasnosti 4 a bude musieť obetovať horalku. Pri pozícií 3 a 4 bude treba obetovať horalku hneď v prvom kole kvôli riešeniu úžasnosti 4 a potom aj kvôli riešeniu úžasnosti 3 (ktoré vyhralo súboj v prvej dvojici). Konkrétne, ak by Romanko začínal na tretej pozícií, na testovači by boli úžasnosti riešení v poradí 1 3 2 4.

6. Odplata drepovaním

kat. O; 12 b za popis, 8 b za program

Byť vedúci na sústreďení je naozaj super. Hlavne preto, že máte na starosti 32 účastníkov, ktorí vás musia poslúchať bez ohľadu na to, čo im prikážete urobiť. A tak sa noví vedúci mstia ďalšej generácii za krivdy, ktoré im boli spôsobené, keď boli ešte účastníkmi. Takto potom tento kolobeh pokračuje donekonečna ...

Emo si tiež živo spomína na príkoria, ktoré podstupoval na sústreďeniach. Najviac nemal rád rozcvičky, lebo ho vedúci vytiahli z postele a nútili behať, skákať a drepovať. Preto sa rozhodol, že teraz donúti účastníkov drepovať on. Vymyslel si na to vskutku zaujímavý spôsob.

Najskôr všetkých účastníkov usporiadal do s zástupov. V každom z nich bolo r účastníkov. Vytvoril tak obdĺžnik veľkosti $r \times s$. Následne si opakovane vyberal riadok alebo stĺpec tohto obdĺžnika a nútil účastníkov v danom riadku alebo stĺpci spraviť poldrep. To znamená, že ak účastník stál, tak si musel čupnúť a ak čupel, tak sa musel postaviť.

Ak napríklad ukázal na tretí stĺpec, všetci účastníci v treťom zástupe si museli čupnúť a ostatní ostali stáť. Ak potom ukázal na druhý riadok, tak všetci účastníci v druhom riadku si museli čupnúť, okrem účastníka, ktorý bol v druhom riadku a treťom stĺpci. Ten totiž už čupel a preto sa musel postaviť.

Keď to Ema prestalo baviť⁹, pustil účastníkov na raňajky. Ešte pred tým si však zapamätal, že na konci čupelo presne k účastníkov. Keď potom prišiel na raňajky, sadol si vedľa Miša, ktorému vysvetlil, čo robil. Miša táto hra zaujala a chcel vedieť, koľkými spôsobmi vedel Emo doceliť stav, pri ktorom na konci čupelo k ľudí. Nevedel si s tým však poradiť, preto poprosil Ema, aby mu povedal ešte niečo viac. Emo mu preto prezradil, že keď si vyberal stĺpce a riadky, ktoré majú spraviť poldrep, riadok si vybral p_r -krát a stĺpec p_s -krát.

To už Mišovi na vyriešenie úlohy stačilo a povedal si, že vám takúto úlohu zadá, aby ste sa potrápili aj vy.

Úloha

Na začiatku mal Emo stojacich účastníkov, ktorí boli rozdelení do obdĺžnika s r riadkami a s stĺpcami. Následne Emo určoval niektoré riadky alebo stĺpce a účastníci v nich museli zmeniť svoj stav – ak stáli, museli si čupnúť, ak čupeli, museli sa postaviť.

Vypočítajte, koľkými spôsobmi mohol určiť p_r riadkov a p_s stĺpcov tak, aby na konci ostalo k čupiacich ľudí. Dve možnosti považujeme za rôzne, ak existuje riadok alebo stĺpec, ktorý bol vybraný v oboch možnostiach rôzny počet krát. Nezáleží nám teda na poradí, v akom Emo riadky a stĺpce určoval.

Keďže výsledok môže byť príliš veľký, vypíšte iba jeho zvyšok po delení číslom 555 555 555.

Formát vstupu

Vstup obsahuje päť medzerou oddelených čísel r, s, p_r, p_s a k – počet riadkov a stĺpcov obdĺžnika, počty vybratí niektorého riadka, počty vybratí niektorého stĺpca a počet účastníkov, ktorí na konci ostali čupieť.

Pre tieto hodnoty platí, že $1 \leq r, s \leq 1\,500$, $0 \leq p_r, p_s \leq 1\,500$ a $0 \leq k \leq rs$.

⁹Po asi hodine a pol.

Naviac, v prvej sade vstupov platí, že $1 \leq r, s, p_r, p_s \leq 10$, v druhej sade vstupov platí, že $1 \leq r, s \leq 20$ a v tretej sade platí, že $1 \leq r, s, p_r, p_s \leq 100$.

Formát výstupu

Vypíšte jedno číslo – počet možností, ktorými vieme určiť p_r riadkov a p_s stĺpcov tak, aby ostalo čipieť k účastníkov. Keďže toto číslo môže byť príliš veľké, vypíšte iba jeho zvyšok po delení číslom 555 555 555.

Príklad

vstup	výstup
2 2 2 2 4	4

V dvoch možnostiach vybral Emo oba riadky a jeden zo stĺpcov dvakrát. V druhých dvoch možnostiach vybral oba stĺpce a jeden z riadkov dvakrát.

vstup	výstup
2 2 0 0 1	0

Ak Emo nevybral žiaden riadok ani stĺpec, nie je možné aby niekto čupel.

vstup	výstup
8 4 12 3 8	27600

7. Obávaný skok slyší, horizontálno-zvislý

kat. O; 12 b za popis, 8 b za program

Počuli ste už o paraglidingu? Je to rekreačno-adrenalinový šport v štýle ‘riadeného voľného pádu’ – inak povedané, po skoku z vyvýšeného miesta sa s pomocou vyčáčkovaného padáka vznášate nad okolitou krajinou, pričom vás gravitácia pomaly, ale neúprosne priťahuje k zemi.

Paragliding, ako každý správny šport, má svoj klub nadšencov s neoriginálnym názvom – Klub Slovenských Paraglidistov (KSP). A ako každý správny zväz organizuje KSP klubové akcie – spoločné výlety do prestížnych paraglidingových centier, kde sa členovia klubu do sýtosti vylietavajú v tých najlukratívnejších pohoriach. Samozrejme, každá akcia musí byť úžasnejšia ako tá predošlá, a preto je výber správnej destinácie kľúčový. Tento rok sa dokonca na akciu prihlásila legenda Tatranského paraglidingu Sysel Samovrah. Ten je svojou vytrvalosťou a zápalom pre tento šport známy na svetovej paraglidingovej scéne.

V jeho bestseller autobiografii sa členovia KSP dočítali, že Sysel si najviac užíva veľké zoskoky – také, kde prevýšenie, ktoré preletí, je väčšie ako m . V rámci exkluzívneho rozhovoru tiež spomenul, že na paraglidingu mu najviac prekáža nutnosť po každom skoku vyšliapať naspäť na kopec. Preto väčšinou skočí z vrchola jedného kopca a pristane na vrchole nejakého nižšieho, z ktorého môže opäť skočiť ďalej. Ak toto urobí $s - 1$ krát, už mu v porovnaní s počtom skokov výstup na prvý kopec až tak neprekáža. No a z odpočutého telefónneho rozhovoru vysvitlo, že Sysel rád lietava po dobrom obede, ale zásadne smerom na východ, keďže pri lete na západ by mu svietilo slnko do očí.

KSP nechce Sysla na svojej akcii sklamať, a preto hľadá také pohorie, ktoré obsahuje najviac trás vyhovujúcich Syslovi. Napíšte program, ktorý im v tom pomôže.

Úloha

Každé pohorie s n horami sa dá opísať ako postupnosť n čísel, reprezentujúcich výšky hôr v ňom, v smere od západu na východ. Trasa je ľubovoľná postupnosť niekoľkých, nie nutne susediacich hôr, pričom Syslovi vyhovuje práve vtedy, ak obsahuje presne s hôr takých, že medzi každou za sebou idúcou dvojicou hôr je prevýšenie väčšie ako m – teda ak výška západnejšej hory je x a výška východnejšej hory je y , tak musí platiť, že $x - y > m$. Pre dané pohorie tvorené n horami a hodnoty m a s nájdite počet trás, ktoré vyhovujú Syslovm požiadavkam.

Formát vstupu

Na prvom riadku sú tri čísla n , m a s ($1 \leq n \leq 10^5$, $0 \leq m \leq 10^{18}$, $2 \leq s \leq 20$) – počet hôr v pohorí, najmenšie prevýšenie ktoré Sysla uspokojí, a počet hôr v trase, ktorá mu vyhovuje.

V druhom riadku je n čísel h_i ($1 \leq h_i \leq 10^{18}$) – výšky hôr v pohorí, od západu na východ.

Navyše pre jednotlivé vstupné sady platí

Číslo sady	1	2,3	4,5,6	7,8
Maximálne n	20	10^3	10^5	10^5
Maximálne m a h_i	10^6	10^6	10^6	10^{18}

Formát výstupu

Vypíšte jedno číslo – počet rôznych trás, ktoré obsahujú presne s hôr a každá hora na trase je o viac ako m vyššia od tej nasledujúcej. Dve trasy považujeme za rôzne, ak existuje aspoň jedna hora, ktorá patrí do jednej z nich, ale nie do druhej.

Kedže toto číslo môže byť príliš veľké, pre KSP stačí vedieť tento počet modulo $1\,000\,000\,007$ ($10^9 + 7$).

Príklad

vstup	výstup
<div style="border: 1px solid black; padding: 5px;">4 0 2 5 4 3 1</div>	<div style="border: 1px solid black; padding: 5px;">6</div>
<i>Systövi vyhovuje ľubovoľná dvojica hôr.</i>	
vstup	výstup
<div style="border: 1px solid black; padding: 5px;">4 1 2 5 4 3 1</div>	<div style="border: 1px solid black; padding: 5px;">4</div>
<i>Systövi už nevyhovujú dvojice (5, 4) a (4, 3)</i>	
vstup	výstup
<div style="border: 1px solid black; padding: 5px;">8 3 3 11 15 15 10 10 7 1 1</div>	<div style="border: 1px solid black; padding: 5px;">14</div>

8. Obmedzený pohyb

kat. O; 12 b za popis, 8 b za program

Farmár Bob sa nedávno rozhodol, že si na svoju farmu dá konečne zaviesť inžinierske siete. Už o pár dní sa mu to na záhrade len tak hemžilo robotníkmi kopajúcimi dlhočizné jamy pre rôzne káble a potrubia. Firma, ktorá túto robotu robila, však nanešťastie skrachovala a Boba nechala s rozkopanou záhradou. Bob sa však po záhrade potrebuje pohybovať, musí sa predsa starať o petržlen, baklažán, cibuľku, jahody ... Kedže skákanie cez jamy je príliš nebezpečné, obchádzať sa mu ich nechce a zasypávanie všetkých jám by trvalo príliš dlho, zobral Bob niekoľko dosák a položil ich krížom cez niektoré jamy ako mosty.

Po čase však Bob zistil, že niektoré z týchto dosák by sa mu zišli pri oprave plota. Rozhodol sa teda niektoré jamy zasypať, aby tým nejaké dosky uvoľnil. Ale ktoré? Zišlo by sa mu pre každú jamu vedieť, ktoré dosky cez ňu vlastne vedú.

Úloha

Na záhrade je n jám očíslovaných $1, 2, \dots, n$ a m dosák očíslovaných $1, 2, \dots, m$. Jamy aj dosky budeme pre jednoduchosť považovať za úsečky v rovine. Žiadne dve jamy nemajú spoločný bod, ani žiadne dve dosky nemajú spoločný bod. Navyše platí, že **každá doska sa pretína s práve jednou jamou** (pod pojmom "pretína sa" myslíme, že úsečky majú práve jeden spoločný bod a tento bod nie je koncovým bodom žiadnej z nich). Na vstupe dostanete zadané pozície všetkých jám aj dosák. Pre každú jamu vypíšte zoznam dosák, ktoré ju pretínajú.

Formát vstupu

V prvom riadku vstupu sú dve celé čísla n, m ($1 \leq n, m \leq 50\,000$) – počet jám a počet dosák. V každom z nasledujúcich n riadkov je popis jednej jamy. Popis jamy tvoria štyri celé čísla x_1, y_1, x_2, y_2 ($-1\,000\,000 \leq x_1, y_1, x_2, y_2 \leq 1\,000\,000$), ktoré popisujú to, že daná jama má koncové body (x_1, y_1) a (x_2, y_2) . V nasledujúcich m riadkoch sú popisy dosák, v rovnakom formáte. Jamy aj dosky sú očíslované v poradí, v akom sú uvedené na vstupe.

Formát výstupu

Pre každú jamu (v poradí, ako sú očíslované), vypíšte jeden riadok. Tento riadok má obsahovať čísla dosák, ktoré túto jamu pretínajú, oddelené **presne jednou medzerou**, vo **vzostupnom poradí**. V prípade, že jamu nepretína žiadna doska, vypíšte pre ňu prázdny riadok.

Príklad

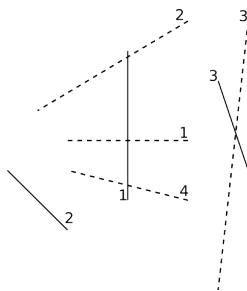
vstup

```
3 4
1 -1 1 4
-1 -2 -3 0
4 3 5 0
3 1 -1 1
3 5 -2 2
5 5 4 -4
3 -1 -1 0
```

výstup

```
1 2 4
3
```

Situácia na Bobovej záhrade vyzerá nasledovne (plné čiary sú jamy, prerušované čiary sú dosky):



Zadania kategórie T

Nezabudnite, že môžete riešiť aj kategóriu T (je trochu ťažšia ako kategória O, ale mnohí z vás ju určite zvládnu).

Body z tejto kategórie sa mierne zohľadňujú aj pri výbere tímu, ktorý pôjde súťažiť na [Medzinárodnú olympiádu v informatike \(IOI\)](#), tento rok v Iráne.