



Korespondenčný seminár z programovania

Leták zimnej časti XXXVIII. ročníka

Korespondenčný seminár z programovania (KSP) je súťaž programátorov – stredoškolákov a mladších – pripravovaná skupinou študentov Fakulty matematiky, fyziky a informatiky Univerzity Komenského v Bratislave. Naším cieľom je zdokonaľiť žiakov v programovaní a v algoritmickom myslení.

Riešením súťažných úloh a štúdiom vzorových riešení sa zlepšíš v programovaní a naučíš sa algoritmicky rozmýšľať. Získané poznatky a skúsenosti využiješ v iných súťažiach v programovaní (napríklad pri riešení [Olympiády v informatike](#)), v bežnom živote, počas vysokoškolského štúdia, dokonca aj na prijímacích pohovoroch do zamestnania. Naši riešitelia sa každoročne zúčastňujú a úspešne umiestňujú na medzinárodných olympiádach v informatike (v Austrálii, Taliansku, Kazachstane, Taiwane, ...). Mnoho našich bývalých riešiteľov sa tiež bez ťažkostí zamestnalo v špičkových IT spoločnostiach ako Google, Facebook, ESET, ...

Ak študuješ na strednej škole a zaujíma ťa programovanie, neváhaj a zapoj sa do KSP:

Ako sa zapojiť do KSP?

- **Prečítaj** si zadania. Nájdeš ich v tomto letáku a na našej stránke <https://www.ksp.sk/ulohy>. Každý rok máme zimnú a letnú časť, obe majú dve kolá s ôsmimi úlohami.
- Teš sa, aké sú tento rok pekné úlohy.
- **Vyrieš** úlohy. Nemusíš vyriešiť všetky, nemusíš ich vyriešiť najlepšie ako sa dá. Aj za čiastočné riešenia sa dostávajú body, za každú úlohu za dá získať 0 až 20 bodov.
- Na riešenie úloh jedného kola máš približne dva mesiace a môžeš ich riešiť doma bez toho, aby si niekam cestoval. Termín odovzdania úloh je napísaný aj na našej stránke, aj v PDF zadaniach. Úlohy sa nedajú odovzdávať po termíne, takže si to, prosím, nenechaj na poslednú chvíľu.
- Úlohy rieš samostatne a neprezrádzaš riešenia ostatným riešiteľom. Odpisovanie riešení a prezradenie riešení pred termínom kola je porušením pravidiel KSP. Po skončení kola sa, samozrejme, o riešeniach rozprávať môžeš. :)
- **Odovzdaj** riešenia úloh. Odkaz na odovzdávanie úloh nájdeš pod webovým zadaním každej úlohy alebo na stránke <https://www.ksp.sk/odovzdavanie>. Na odovzdávanie sa treba prihlásiť, aby sme vedeli, komu máme dať body.
 - Vo väčšine úloh odovzdávaš program a popis.
 - Program je hneď po odovzdaní otestovaný testovačom a hneď vidíš, koľko bodov za program máš. Program môžeš odovzdávať znova a znova, až kým nie si spokojný s výsledkom. Ak nevieš, ako majú vyzeráť odovzdané programy, pozri si <https://www.ksp.sk/odovzdavanie-programov>
 - Do popisu slovne napíšeš, ako tvoje riešenie funguje, prečo funguje a tiež odhad časovej a pamäťovej zložitosti programu. Viac sa dozvieš na stránke <https://www.ksp.sk/ako-riesit>. Popis opraví a obodujú vedúci KSP po skončení kola.
- Po skončení kola si **prečítaj vzorové riešenia** úloh (veľa sa z toho naučíš), pozri svoje opravené popisy (či ti tam vedúci nenapísali nejaké poučné komentáre), pozri sa do výsledkovky a **teš sa**, koľko máš bodov. Vo výsledkoch sa hodnotí samostatne letná a zimná časť. V každej časti je dôležitý celkový súčet bodov.
- Prečo sa máš tešiť z bodov? Čítaj ďalej.

Čo môžem vyhrať?

- Okrem neoceniteľných vedomostí, skúseností a zručností, ktoré získaš pri riešení semináru, môžeš vyhrať množstvo skvelých vecí.
- Všetci víťazi od nás dostanú **vecné ceny**.
- Pre 36 najlepších riešiteľov organizujeme každoročne dve týždenné **sústredenia**. Sústredenie je niečo ako tábor, na ktorom spoznáš nových priateľov s podobnými záujmami, naučíš sa čosi viac nielen o programovaní a zažiješ kopec zábavy. Sústredenia sú fakt skvelé akcie, najmä, keď ich organizuje Trojsten.

- Aby ste sa mohli pochváliť ostatným, akí ste šikovní, víťazom všetkých levelov udelíme a pošleme **diplomy**.
- Aj keď sa nedostaneš medzi víťazov, stále môžeš byť úspešným riešiteľom. Úspešný riešiteľ je ten, kto získal aspoň polovicu bodov počas celej časti (letnej, či zimnej).

Pravidlá a levely

Počnúc tridsiatym piatym ročníkom rušíme staré kategórie a prechádzame na nový systém *levelov*.

Každý riešiteľ má level, číslo od 1 po 4. Noví riešitelia začínajú na leveli 1 a pokiaľ sa im v riešení darí, level im postupne rastie. Svoj level si môže každý riešiteľ pozrieť na našej stránke. Riešiteľom s levelom L sa započítavajú body len za úlohy s číslami L až 8.

Vo výsledkových listinách (<https://www.ksp.sk/vysledky>) sa každému riešiteľovi počíta **5 najlepšie vyriešených úloh**. Celkovo sa dá za časť (dve kolá) získať 200 bodov. Riešitelia, ktorí sa v nejakej výsledkovke umiestnili na jednom z prvých dvoch miest a majú aspoň 150 bodov sú **víťazi**. Najlepších 36 riešiteľov pozývame na sústredenie.

Podrobnejšie pravidlá si môžete prečítať na <https://www.ksp.sk/pravidla>.

Registrácia

Pred odovzdaním riešenia je potrebné sa zaregistrovať na našej webstránke a vyplniť požadované kontaktné údaje. Odporúčame sa zaregistrovať aspoň pár dní pred odovzdávaním riešenia (pre prípad, že by ste mali počas registrácie nejaké problémy).

Účasťou v KSP nám dávate súhlas spracovať a archivovať údaje, ktoré nám poskytnete pri registrácii, ako aj zverejniť vaše meno, školu, ročník a získané body vo výsledkovej listine.



Úlohy 1. kola zimnej časti

Termín odoslania riešení tohto kola je pondelok **2. novembra 2020**. Doprogramovávanie končí v podnelok 16. novembra 2020.

1. Hádam prežijeme zimu

12 b za popis, 8 b za program

V lese sa už stmieva a hovniválky Timka a Gabika sa vracajú pomaly domov. Ide im to pomaly, lebo každá si pred sebou tlačí dnešný úlovok, guľku trusu. Keď prídu ku svojej skrýši, starostlivo pridajú novú guľku do zásob na zimu. Obe majú pekne v jednom rade naskladaných pred svojou chodbou už mnoho kôpok gulôčok. Vedia, že sa blíži zima a nebudú sa tak často vídavať, preto Timka navrhla, že si navzájom darujú jednu kôpku guliek, aby na seba nezabudli. Gabika súhlasila s jednou podmienkou. Každá musí mať po tejto výmene dokopy vo všetkých svojich kôpkach párnny počet gulôčok, aby si vedela zásoby rovnomerne rozdeliť na prvú a druhú polovicu zimy. Ak sa táto podmienka nedá splniť, k výmene nedôjde.

Úloha

Na vstupe dostanete samostatne popis Gabikíných a Timkíných kôpok gulôčok. Oba z nich zapísané pomocou kladných celých čísel. Každé číslo hovorí, koľko guliek je na danej kôpke. Gabika a Timka majú rovnako veľa kôpok a navzájom si vymenia práve jednu kôpku guliek, ak každá bude mať po výmene v súčte na všetkých svojich kôpkach párnny počet gulôčok. Vašou úlohou je povedať, či dôjde k výmene alebo nie.

Formát vstupu

Na prvom riadku vstupu je celé číslo n z rozsahu od 1 po 100 000, počet kôpok jednej hovniválky. Na druhom riadku je n celých čísel z rozsahu 1 až 10 000, popis Gabikíných kôpok a na treťom riadku sa nachádza n celých čísel z rozsahu 1 až 10 000, popis Timkíných kôpok. Na druhom a treťom riadku vždy i -te číslo v riadku hovorí, koľko guliek je na i -tej kôpke danej hovniválky.

Formát výstupu

Na jeden riadok výstupu vypíšete **ano**, ak k výmene dôjde alebo **nie**, ak k výmene nedôjde.

Hodnotenie

Je 8 sád vstupov. Platia v nich nasledujúce obmedzenia:

Sada	1–2	3–4	5–8
$1 \leq n \leq$	100	1 000	100 000

Príklad

vstup

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

výstup

```
nie
```

Nech si vymenia ktorúkoľvek dvojicu kôpok, nebudú mať po tejto výmene obe párnny počet všetkých svojich gulôčok.

vstup

```
3
2 8 64
57 20 3
```

výstup

```
ano
```

V tomto prípade k výmene dôjde, napríklad môže dať Gabika Timke kôpku s 8-mimi guľkami a Timka Gabike kôpku s 20-timi guľkami. Gabika bude mať po tejto výmene $2 + 20 + 64 = 86$ guliek a Timka $57 + 8 + 3 = 68$,

teda obe majú párny počet guliek.

2. Oválne mince

12 b za popis, 8 b za program

V meste hovniválov sa deje veľká udalosť – ide sa stavať obrovský palác z trusových gulôčok. Stavajú sa na počesť Hovnivála I., ktorý zjednotil všetky kmene hovniválov do jednej veľkej ríše a priniesol medzi nich pokoj a mier. Keď sa zhromažďovali gulôčky, bolo jasné, že ho nebude také ľahké postaviť. Začal sa konkurz na stavbu paláca. Tento konkurz vyhrala firma, kde pracujú Janko a Ferko, a práve oni dostali poverenie postaviť tento palác. Ako roky plynuli, stavbári čakali už len na jedno – peniaze za túto neľahkú stavbu. A ten čas nastal dnes.

Kto vykonal viac práce? Janko či Ferko? Boli ste poverení vyplatiť im ich zaslúžené peniaze. Ale pozor! Nechcete, aby sa ich výplaty veľmi odlišovali, inak sa pobijú!

Úloha

Máme n mincí s hodnotami 1 až n . Keďže chceme, aby sa nepobili, musíme rozdeliť tieto mince čo najviac rovnomerne obidvom stavbárom. Každú mincu máte k dispozícii len raz. Pochopiteľne ju musíte dať len jednému stavbárovi. Vašou úlohou je nájsť čo najmenší rozdiel výplat Janka a Ferka.

Formát vstupu

Na prvom riadku vstupu dostanete číslo t , $1 \leq t \leq 1000$, počet výplat, ktoré chcete rozdeliť. Na každom z ďalších t riadkov sa nachádzajú dve medzery oddelené čísla n_i a p , kde číslo n_i znamená počet mincí, ktoré máme pre danú výplatu (mince majú teda hodnoty $1, 2, \dots, n_i$), a $p \in \{0, 1\}$. Platí, že $1 \leq n_i \leq 3000$

Formát výstupu

Ak sa $p = 0$, na výstup vypíšete pre danú výplatu jedno číslo na samostatnom riadku – minimálny rozdiel výplat Janka a Ferka.

Ak sa $p = 1$, na výstup pre danú výplatu okrem jedného čísla na samostatnom riadku – minimálneho rozdielu výplat, vypíšete aj príklad na nejaké rozdelenie mincí, kde je rozdiel výplat minimálny. Pre každú výplatu okrem minimálneho rozdielu vypíšete ďalšie dva riadky.

Prvý z nich na začiatku obsahuje číslo j – počet mincí, ktoré dostane Janko. V tom istom riadku nasleduje medzera a j medzerou oddelených čísel – hodnoty Jankových mincí.

Druhý z nich na začiatku obsahuje číslo f – počet mincí, ktoré dostane Ferko. V tom istom riadku nasleduje medzera a f medzerou oddelených čísel – hodnoty Ferkových mincí.

Obmedzenia

V prvej štvrtine sád platí $p = 0$.

Poznámka

V posledných sádach je treba vypisovať pomerne veľa čísel, a v prípade, ak to robíte pomaly (a najmä v pomalom programovacom jazyku ako napr. Pythone), vám môže byť neakceptované riešenie, ktoré má vzorovú časovú zložitosť. Na vypisovanie poľa odporúčame namiesto:

Listing programu (Python)

```
for i in range(len(pole)-1):
    print(pole[i], end=' ')
print(pole[-1])
```

použiť

Listing programu (Python)

```
print(" ".join(str(prvok) for prvok in pole))
```

Príklad

vstup

```
2
3 0
5 0
```

výstup

```
0
1
```

V prvom prípade máme mince hodnôt 1, 2, 3. Medzi dvoch ľudí ich vieme rozdeliť tak, že prvý dostane mince 1 a 2 a druhý dostane mincu 3. Rozdiel medzi súčtami ich mincí je teda 0. V druhom prípade máme mince 1, 2, 3, 4, 5. Tieto mince vieme najlepšie rozdeliť tak, že prvý dostane mince 1, 2, 5 a druhý mince 3, 4, teda rozdiel medzi súčtami ich mincí je 1.

vstup

```
2
3 1
5 1
```

výstup

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

Iný príklad na 5 mincí môže byť napríklad aj tak, že prvý človek dostane mince 1, 2, 4 a druhý 3, 5

3. Vystupovanie

12 b za popis, 8 b za program

Každý víkend sa Jozef Hovnivál potreboval dostať domov z rodinného výmetu, aby si mohol sadnúť na svoju pohodlnú stolicu. Chodieval autotrusom. No aby to nemal také jednoduché, zakaždým sa rozhodol, akú dlhú prechádzku si chce spraviť domov z autotrusovej zastávky. Teda vystúpil na takej zastávke, ktorá je od jeho domu v správnej vzdialenosti. No toto ho rýchlo omrzelo, pretože chodil často tou istou trusou. Tak si vymyslel ešte jednu podmienku, a to že nepôjde z ktorejkoľvek zastávky, ktorá je v správnej vzdialenosti, ale z k -tej takej v poradí. Teda napríklad sa mohol rozhodnúť, že vystúpi na tretej zastávke, z tých, ktoré sú od jeho domu vzdialené 7 metrov. Jozefovi Hovniválovi ale zaberalo veľmi veľa času zistiť, na ktorej zastávke má vystúpiť. Preto potrebuje vašu pomoc.

Úloha

Na vstupe dostanete n čísel v rozsahu od 1 po 1 000 000. Toto sú vzdialenosti jednotlivých zastávok od jeho domu, v takom poradí, v akom ich prejde autotrus. Potom dostanete q otázok. Každá otázka pozostáva z čísla zastávky l , na ktorej Jozef nastúpi, čísla zastávky r , na ktorej mu skončí platnosť lístka (a teda nemôže pokračovať ďalej v ceste autotrusom). Ďalej dostanete dĺžku prechádzky v , ktorú si Jozef praje a nakoniec na kolke takej zastávke chce Jozef vystúpiť, k . Vašou úlohou je vypísať číslo zastávky, ktorá sa nachádza v intervale od l po r vrátane, jej vzdialenosť od domu je v a je to k -ta taká zastávka v danom intervale. Ak taká neexistuje, vypíšte -1 . Číslujeme od jednotky.

Formát vstupu

Na prvom riadku vstupu sa nachádzajú čísla n a q oddelené medzerou, počet zastávok a počet otázok. Na druhom riadku sa nachádza n čísel oddelených medzerami, vzdialenosti zastávok od Jozefovho domu. Nasleduje q riadkov, na každom z ktorých sa nachádzajú medzerami oddelené čísla l , r , v a k , ktorých význam je vysvetlený vyššie.

Formát výstupu

Vypíšte q riadkov výstupu. Na i -tom riadku sa nachádza odpoveď na i -tu otázku, teda číslo zastávky, ktorá spĺňa požadované vlastnosti, alebo -1 ak taká neexistuje.

Hodnotenie

Je 8 sád vstupov. Platia v nich nasledujúce obmedzenia:

Sada	1–2	3	4	5–8
$1 \leq n \leq$	100	10 000	80 000	100 000
$1 \leq q \leq$	100	10 000	100 000	100 000

V prvej sade navyše platí, že vzdialenosti zastávok od domu sú z množiny $\{1, 2\}$.

Príklad

vstup

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

výstup

```
2
-1
2
-1
```

vstup

```
10 5
3 3 1 3 3 2 1 3 3 1
1 5 3 1
1 8 3 1
2 8 3 2
1 4 1 1
1 6 1 2
```

výstup

```
1
1
4
3
-1
```

4. Nové PINy pre trezor

12 b za popis, 8 b za program

Bola jar. Sniežik sa topil, kvietky kvitli, rástli listy. Ale najmä, hovnivále kládli vajíčka. Ale to nie je len tak, naklást' vajíčka a bezstarostne sa zabávať. Treba ich schovať, ukryť pred beštiami. Náš Lajniak Lajko to mal tentokrát obzvlášť obtiažne. Jeho partnerka ho požiadala, aby zohnal trus, v ktorom pod zemou schovávajú vajíčka. Tento materiál je nevyhnutný, aby sa larvy mali čím živiť.

Lajko je ale veľmi zábudlivý lajniak, a minulé leto zabudol svoje zásoby trusu vo Firme Korporátnych Slimákov, kde v tom čase brigádoval. Aby ich dostal naspäť, musel by otvoriť svoj starý firemný trezor. Bohužiaľ, zabudol aj svoj PIN. Našťastie, FKS ponúka zmenu PINu, ak si ho niekto zabudol. Tento proces funguje nasledovne:

Firma Korporátnych Slimákov Lajkovi zaradom navrhne p rôznych PINov, každý o rovnakej dĺžke n . Užívateľ ide zaradom, a keď je na PINE, ktorý sa mu páči, môže si ho zvoliť. Systém ale ponúka aj ďalšiu zmenu, ak si užívateľ chce výber rozmyslieť. Má to ale háčik. Môže si ďalší PIN zvoliť iba ak sa líši v len jednej cifre od toho, ktorý si zvolil naposledy. Takto si užívateľ môže PIN zmeniť koľkokrát len chce, pokiaľ sa každý ďalší PIN líši v len jednej cifre. Keďže tento systém naprogramovali slimáci, zmena PINu prebieha pomocou operácie, ktorá zmení jednu cifru v PINE o 1, buď sa cifra zväčší, alebo zmenší. Keby si Lajko chcel zmeniť PIN 0000 na PIN 0700, systému by to trvalo 7 operácií.

Lajko ako minulý brigádnik vo firme o tomto systéme vie. Je zároveň veľmi citlivý, čo sa týka problémov jeho pamäte. Chce sa teda vyhnúť zmene PINu. Čiže samozrejme sa namiesto toho pokúsi prefažiť slimačí systém menenia PINu tým, že ho donúti vykonať čo najviac operácií pre každú sadu navrhnutých PINov. Keď systém prefaží, určite sa mu trezor otvorí, a dostane sa k svojmu vysnívanému trusu. Pomôžte Lajkovi zistiť, koľko najviac operácií vie od systému v danej sade vynútiť.

Úloha

Dostanete p rôznych PINov, každý o rovnakej dĺžke n . V PINE môžu byť aj nuly. Jeden PIN si zvolíte ako začiatočný. Následne si môžete vždy PIN zmeniť na ľubovoľný v zozname nasledujúci PIN, pokiaľ sa od toho, čo máte teraz, líši len v jednej cifre. Ak sa líši vo viacerých cifrách, zobrať si ho nemôžete a idete na ďalší navrhnutý PIN. Keď si takto zmeníte PIN, tak podľa toho, o koľko sa dve rozdielne cifry líšia, toľko sa vykoná operácií. Keď sa dostanete na posledný PIN, po ňom si už ďalej PIN meniť nemôžete.

Vašou úlohou bude zistiť, koľko najviac operácií sa dá vykonať pre daný zoznam navrhnutých PINov.

Formát vstupu

Na prvom riadku dostanete číslo n ($1 \leq n \leq 6$) – počet cifier v každom PINE. Na druhom riadku bude číslo p ($1 \leq p \leq 100\,000$) – počet PINov.

Nasleduje p riadkov, na i -tom sa nachádza PIN p_i , pričom PIN je ľubovoľná kombinácia cifier 0 až 9 o dĺžke n .

Formát výstupu

Vypíšte jedno číslo – najväčší možný počet operácií, ktoré sa dajú vykonať.

Príklad

vstup

```
4
6
8823
2145
2185
3385
4145
4445
```

výstup

```
5
```

Lajko si zvolí 2145 ako začiatocný PIN. Od neho sa môže dostať na 2185, čo by vykonalo 4 operácie, alebo na 4145, čo by vykonalo iba 2. Lenže sa mu oplatí zvoliť si 4145 aj tak, pretože z neho sa vie dostať na 4445, čo by dokopy vykonalo 5 operácií. Neexistuje iná stratégia, pomocou ktorej by systém vykonal viac, ako 5 operácií.

vstup

```
5
11
00000
60000
00100
10100
60100
10190
10390
61100
20100
10380
20190
```

výstup

```
20
```

5. Iba aby hladný nebol

12 b za popis, 8 b za program

Neďaleho hovniválovho bytu je park, v ktorom je síce dosť chodníkov na to, aby sa dalo dostať z každého na každý, ale netvorí žiaden cyklus. Hovniválovi sa teda určite nestane, že by chodil dokola mysliac si, že už o chvíľu určite príde na chodník s lahodnou pochutinou.

Taký park by ale bez neporiadnych psíčkarov vôbec nebol lákavý. Kde-tu sa vyskytne mäkká, hnedastá kôpka maškrtky. Jednoducho raj pre hovniválov. Ten náš má vymyslenú vynikajúcu stratégiu. Každú križovatku chodníkov si očísloval. Vždy, keď ide do parku zháňať niečo pod zub, rozhodne sa, ktorých k križovatiek postupne navštívi. Tieto križovatky nemusia byť priamo spojené chodníkom. Medzi dvoma križovatkami sa vždy presúva najkratšou možnou cestou. Ako tak kráča po chodníkoch, dúfa, že aspoň na jednom z nich nájde voňavú pochúťku.

Hovnivála by zaujímalo, koľko rôznych prechádzok, na ktorých nájde aspoň nejakú tú podslinku, si vie naplánovať.

Úloha

Poznáte popis parku. O každej z n križovatiek viete, ktoré z čísel od 1 do n jej hovnivál priradil. Každé dve križovatky majú rôzne čísla.

Križovatky sú rôzne pospájané chodníkmi. Chodníkov je $n - 1$ a platí, že sa po nich dá dostať z každej križovatky do každej inej.

Na každom chodníku sa buď nachádza, alebo nenachádza exkrement.

Prechádzka dĺžky k je postupnosť k križovatiek. Križovatky v postupnosti **nemusia** byť priamo spojené hranou. Dokonca môže tá istá križovátka byť v postupnosti viackrát za sebou.

Hovnivál si vždy najskôr vyberie prechádzku dĺžky k a následne sa medzi križovatkami vo zvolenej prechádzke presúva najkratšou možnou cestou.

Vašou úlohou je zistiť, koľko rôznych prechádzok dĺžky k si môže hovnivál zvoliť, aby pre každú z nich platilo, že počas nej prejde po aspoň jednom chodníku s exkrementom.

Formát vstupu

Na prvom riadku sa nachádzajú čísla n – počet križovatiek a k – dĺžka prechádzky.

Nasleduje $n - 1$ riadkov. V každom z nich sa nachádzajú čísla a, b, x . Tieto čísla hovoria, že existuje chodník medzi križovatkami a a b a exkrement sa na ňom nachádza, ak $x = 1$ a nenachádza, ak $x = 0$.

Formát výstupu

Na výstup vypíšete jedno číslo – počet rôznych prechádzok dĺžky k , počas ktorých hovnivál prejde po aspoň jednom chodníku s exkrementom modulo $10^9 + 7$.

Hodnotenie

Sú 4 sady vstupov. Platia v nich nasledovné obmedzenia:

Sada	1	2-3	4
$2 \leq n \leq$	10	100 000	100 000
$2 \leq k \leq$	5	10	1 000 000

Príklad

vstup

5	4	
1	3	0
2	3	0
3	4	1
4	5	0

výstup

528

Vhodnými prechádzkami sú tu napríklad $[1, 5, 3, 2]$, alebo $[3, 3, 5, 3]$. Nevhodnou je napríklad $[1, 3, 2, 1]$.

vstup

3	3	
1	2	0
3	2	0

výstup

0

Žiadna prechádzka nevyužíva chodník s exkrementom, pretože v celom parku žiadny taký chodník nie je.

6. Váľanie hypergúl

12 b za popis, 8 b za program

“Mohol by si ísť kúpiť teplé rožky do obchodu pls?” dožadovala sa matka, “Rada by som čoskoro konzumovala raňajky!” “Ach jaj,” povzdychol som si, ale nie nahlas, iba v myslí. Nadmieru ma kokoší, keď ma niekto vyrušuje počas ranného online Krúžku Sústredeného Pozerania Tiktoku.

Cesta do obchodu nie je dlhá, ale tak či tak sa ju oplatí skrátiť crossom cez lúky a iné trávnaté plochy. Každých ušetrených 10 sekúnd je jeden pozretý BimBam navyše. Brodím sa vysokým porastom, ale čo to? “Nebodaj je to Scarabeus Multidimensionalis?!” zhíkneť pošepky. Tento druh chrobáčika intímne poznám, totiž, pozeral som o ňom 30 minútový dokumentárny film. Dospelé jedince tejto živočíšnej triedy si vaľkajú vlhké guľky potravy do ktorých nakladú vajíčka, z ktorých sa vyľahnu larvy. Tento konkrétny model hlavohrudca však navyše využíva obranný mechanizmus viacdimenzionálnych hyperguliek na zmätanie predátorov. V tomto prípade to má však nežiadany efekt vyvolania extrémneho záujmu z mojej strany.

Keď som sa znova uvedomil, všimol som si, že sa trochu stmieva. Zdá sa, že som vcelku veľkú časť dňa strávil podrobným skúmaním týchto nezvyčajných guľôčok. Asi už nebudú mať teplé rožky. Nebol to však premárnený čas! Všimol som si, že vnútro guliek prekypuje životom. Vyľahnuté larvy sa v nich hýbu hore-dole v stovkách dimenzií. Častokrát sa stane, že sa vydajú na obrovské dobrodružstvo len aby nakoniec skončili na tom istom mieste, kde začali. Kvôli rokom súťaží, sústredením a iné, mi z mysle reflexívne vyšplechla otázka: “Koľko takýchto rôznych bezvýznamných dobrodružstiev môžu absolvovať?”

Jóóó, ale už ma rozbolel čobrídik a nechcem zmeškať večerníček,¹ tak čo keby ste na to odpovedali **vy**?

Úloha

Na lúke som stretol veľa guliek rôznych dimenzionalít, každú z nich som nejaký čas sledoval. V guľke je larva, ktorá sa neustále hýbe – každú sekundu sa pomotká v smere jednej dimenzie o jednotku vzdialenosti. Guľky sú v porovnaní s larvami omnoho väčšie, takže sa nikdy nestane že by sa prekopali ku okraju.

¹<https://youtu.be/TT7iw3TzEjg>

Pozíciu v D -dimenzionálnom priestore vieme zapísať ako zoznam D súradníc, teda D celých čísel. Larvy sú jednoduché tvory, teda sa iba rozhodnú kam sa chcú pohnúť a následne sa jednu sekundu hýbu rovnakou rýchlosťou rovnakým smerom v práve jednej dimenzii. Potom zastanú, hlboko sa zamyslia a proces opakujú. Za jednu sekundu sa teda vedia dostať do miest, ktoré sa od ich aktuálnej pozície líšia v práve jednej súradnici o práve 1.

Pre každú guľku ma zaujíma, koľkými rôznymi spôsobmi sa vie larva za stanovený čas prevandrovať na miesto, z ktorého začala.

Formát vstupu

Na vstup vstúpi počet guľiek na lúke G , následne nasleduje G riadkov, na i -tom riadku dve čísla – počet dimenzií d_i a počet sekúnd c_i .

Formát výstupu

Na výstup vypíšete G riadkov, na i -tom riadku vypíšete odpoveď pre i -tu guľku modulo $10^9 + 7$.

Obmedzenia

Vstupy budú rozdelené do 4 sád. Každá sada bude hodnotená 2 bodmi. Platia v nich nasledujúce obmedzenia:

Sada	1	2	3	4
$0 \leq G \leq$	10	100	1000	10 000
$1 \leq d_i \leq$	7	50	50	200
$0 \leq c_i \leq$	7	15	50	200

Príklad

vstup

```
4
1 0
1 2
2 1
3 4
```

výstup

```
1
2
0
90
```

7. Áááá, záplavy

12 b za popis, 8 b za program

Kde bolo tam bolo, v pradávnom Egypte bola sieť n hrobiek posvätných hovniválov prepojená $n - 1$ cestičkami. Celý komplex mal krásnu stromovú štruktúru.

Ale táto idylka bola každoročne narušovaná masívnymi záplavami. Záplavy zaplavujú všetko okolo, vrátane hrobiek posvätných hovniválov. Hlavný architekt Dano prišiel za faraónom s famóznou myšlienkou. Čo keby konečne vybudovali tie protipovodňové zábrany z Egyptofondov o ktorých sa už tak dlho hovorí?

Faraón sa poradil so všetkými svojimi kňazmi, a rozhodol, že vôľa vyššej moci je taká, že protipovodňové zábrany vybudujú, ale musia mať tvar obdĺžnika. Navyše, istý hlavný správca hrobiek Čížoš vyhlásil, že keďže treba šetriť, nemôžu postaviť viac ako dve protipovodňové zábrany.

Akoby situácia nebola už príliš komplikovaná, kancelár správy posvätných hovniválov prišiel s požiadavkou, aby boli ochránené hrobky dostupné práve tak, aby sa dalo chodiť na púť a nestratiť sa. Na púť sa vždy ide od nejakej hrobky do inej po predpripravených cestičkách. Aby sa teda pútnici nestratili, navrhoval kancelár mať v dvoch obdĺžnikoch výhradne hrobky na trati púte.

Architekt Dano sa tak chystal pustiť sa do tej namáhavej práce navrhovania, kadiaľ má ísť bariéra, keď zistil že nevie ktorou trasou má vlastne ísť tohtoročná púť. A nebol v tom sám – netušil to ani faraón, ani kňazi, ani Čížoš, ani kancelár. Vlastne jediné, čo sa dozvedel, bolo, že vhodná trasa púte sa niekomu v dostatočnom predstihu prisnil! No to by bolo! Teraz musí vymyslieť, ako preusporiadať hrobky, aby vedel na akúkoľvek trasu púte postaviť protipovodňové ochrany.

Pomôžte mu!

Úloha

Toto je interaktívna úloha. Má dve časti.

Existuje n hrobiek posvätných hovniválov a $n-1$ cestičiek (ktoré musia existovať aj po premiestnení hrobiek) tvoriacich strom.

Pohrebisko je štvorcová pláň s rozmermi $n \times n$ a hrobky môžu byť presunuté iba na celočíselné súradnice.

V prvej časti úlohy dostanete popis cestičiek a váš program vypíše rozloženie hrobiek v rovine.

Po postavení pohrebiska sa viacerým rôznym kňazom snívalo, ktorá púť by bola vhodná. Pre každého zistite, kde by mali byť protipovodňové zábrany postavené. Pochopiteľne, dve obdĺžnikové zábrany by sa nemali prekrývať.

Formát vstupu a výstupu

Na prvom riadku vstupu je číslo n – počet hrobiek, $n \leq 50\,000$.

Na ďalších $n-1$ riadkoch je popis cestičiek: na i -tom riadku sú dve čísla oddelené medzerou, a_i a b_i , $1 \leq a_i, b_i \leq n$, $a_i \neq b_i$.

Následne váš program má vypísať n riadkov, na každom riadku dve celé čísla, x_i a y_i oddelené medzerou. $0 < x_i, y_i \leq n$. Na i -tom riadku výstupu by mali byť súradnice, kde postaviť i -tu hrobku.

Následne na vstupe dostanete niekoľko popisov vysnených pútí. Úloha je **online**, ďalší popis púte dostanete až po úspešnom vypísaní barikád pre predchádzajúcu.

Keď dostanete na vstupe nulu, znamená to, že ste na všetko správne odpovedali a máte ukončiť program.

Každý sen pozostáva z dvoch čísel hrobiek, v ktorých začína a končí púť.

Odpoveď na každú požiadavku vypíšete v nasledovnom formáte:

V prvom riadku vypíšete číslo k ($1 \leq k \leq 2$) – počet protipovodňových zábran. Následne vypíšete k riadkov, v každom štyri čísla: x_1, y_1, x_2 a y_2 – súradnice ľavého dolného a pravého horného rohu (v tomto poradí). ($1 \leq x_1 \leq x_2 \leq n$ a $1 \leq y_1 \leq y_2 \leq n$).

Ak ste odpovedali správne, na vstupe dostanete popis ďalšieho sna (alebo 0 označujúcu ukončenie testovania). Ak nie, testovač vypíše -1. Ak sa váš program vtedy neukončí, hrozí, že miesto WA dostanete TLE alebo EXC.

Aby checker fungoval, ako má, **je nutné**, aby sa po vypísaní pozícií hrobiek a obdĺžnikov výstup presunul z pamäte na štandardný výstup pomocou príkazu `cout.flush()` v C++ alebo `sys.stdout.flush()` v Pythone, pre iné jazyky si vyhľadajte.

Hodnotenie

Sú 4 sady vstupov. V polovici z nich tvoria cestičky binárny strom (z každej hrobky idú najviac tri cestičky).

Počet snov v žiadnej sade nepresiahne 100 000.

V prvých dvoch sadách platí, že aj počet hrobiek, aj počet snov je najviac 1000.

Príklad

vstup	výstup
3 // prvá časť vstupu	1 1 // pozície hrobiek
1 2	1 2
2 3	2 2
1 1 // prvá púť	1 // odpoveď na prvú otázku
1 3 // druhá púť	1 1 1 1
0 // ukončenie vstupu	2 // odpoveď na druhú otázku
	1 1 1 2
	2 2 2 2

Prvá púť pokrýva len jednu hrobku. Stačí jedna protipovodňová zábrana, zakrývajúca presne hrobku číslo 1. V druhom prípade môžeme napríklad postaviť prvú zábranu obklopujúcu prvé dve hrobky a druhú obklopujúcu hrobku 3. Znaký `//` sa v skutočnom vstupe a výstupe neobjavia, ani vy ich nevypisujte. Slúžia len na ilustráciu interakcie

8. Lajno nazmar nevyjde

12 b za popis, 8 b za program

Leto bolo upršané, rastlinám sa darilo. Zvieratá mali čo pod zub, hovnivály kotúľali. Toľko toho nakotúľali, až nevedia, čo s tým. Je im to ľúto len tak všetko prebytočné spláchnuť do záchoda, alebo nechať do ďalšieho leta splesniviť. Zhodli sa teda, že potrebujú mrazák. Nie ale iba taký obyčajný. Obrovský. Mnohospodový.

Rôzne poschodia ale vyžadujú rôzne veľa času na stavbu. Totiž, v niektorých výškach viac fúka, inde je väčšia zima. Hovniválka Elka letným pohľadom do vzduchu odhadla, koľko času bude potrebného na postavenie ktorého poschodia mrazáku. Konkrétne zistila, že na postavenie i -teho poschodia bude potrebného c_i času.

Hovniválka Elka vie, že ak jeden hovnivál buduje guľičku m minút, tisíc hovniválov postaví guľičku za $m/1000$ minút. To, samozrejme, platí aj o stavbe mrazáku. Ak na i -tom poschodí bude pracovať h_i hovniválov, tak toto poschodie dokončia za c_i/h_i času.

Keďže hovnivál má iba také tenké nožičky, rýchlo sa mu unavia. Každý teda zvládne pomôcť postaviť iba jedno poschodie mrazáku. Poschodie sa ale nedá začať stavať, kým nie je dostavané to predchádzajúce, to predsa dá rozum. Hovniválka Elka chce každého z H hovniválov poslať na práve jedno poschodie. Chce to spraviť tak, aby bol celkový čas stavania mrazáku čo najkratší.

Ako najrýchlejšie vedia hovnivály postaviť mrazák?

Úloha

Máme H hovniválov, ktorí chcú postaviť n -poschodový mrazák. Hovniválka Elka chce priradiť každému z H hovniválov nejaké poschodie, na ktorom bude pomáhať so stavbou.

Na každom poschodí, samozrejme, musí pracovať aspoň jeden hovnivál.

Ak je na i -tom poschodí h_i hovniválov, stavba i -teho poschodia zaberie c_i/h_i času.

Vašou úlohou je zistiť, ako najrýchlejšie vie byť mrazák postavený.

Formát vstupu

Na prvom riadku vstupu sú čísla n – počet poschodí mrazáku a H – počet hovniválov. Platí, že $n \leq H$.

Nasleduje n riadkov. Na i -tom z nich je číslo c_i – základný čas potrebný na stavbu i -teho poschodia. Platí, že $1 \leq c_i \leq 100\,000$.

Formát výstupu

Vypíšete jedno číslo – najkratší čas, za ktorý je možné mrazák postaviť.

Vaša odpoveď bude uznaná, ak sa od správnej odpovede nebude odlišovať o viac, ako 1.

Hodnotenie

Sú 3 sady vstupov. Platia v nich nasledovné obmedzenia:

Sada	1	2	3, 4
$1 \leq n \leq$	8	100 000	100 000
$1 \leq H \leq$	20	100 000	10^{12}

Príklad

vstup

```
3 7
6
2
8
```

výstup

```
6.667
```