



Korešpondenčný seminár z programovania

Leták zimnej časti XXXVIII. ročníka

Korešpondenčný 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 neprehrádzaj riešenia ostatným riešiteľom. Odpisovanie riešení a prezradenie riešení pred termínom kola je porušením pravidiel KSP. Po skončení kola sa, samozrejme, o riešeníach rozprávať môžeš. :)
- **Odozdaj** 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 pamätovej 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**. Sústreďenie je niečo ako tábor, na ktorom spoznáš nových priateľov s podobnými záujmami, naučíš sa čosi viac nielen o programovaní a zažiješ kopec zábavy. Sústreďenia 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).

Účastou 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 2. kola zimnej časti

Termín odoslania riešení tohto kola je pondelok **1. februára 2021**. Doprogramovanie končí v pondelok 22. februára 2021.

1. Hady. Prečo to musia byť zrovna hady!?

12 b za popis, 8 b za program

Adam si chce pozrieť nový diel zo seriálu Mandalorian. Adam je fajšmeker a chcel by si to pozrieť v najvyššej 4K kvalite, pre najlepší zážitok. Dokonca si kúpil aj novú televíziu, špeciálne kvôli tomuto seriálu (on Vám povie, že nie, ale neverte mu). Hodí sa do gauča, zapne Disney+ a pozerá sa. Nepozerá sa. Načítava sa.

Streamovať Mandalorian v 4K kvalite vyžaduje veľmi rýchly internet. Adam má dostatočne veľmi rýchly internet, ale nanešťastie, vo svojej sieti použil klasické, obojsmerné sieťové káble. Tieto káble sú ale zastarané a nie dost rýchle na 4K kvalitu. Adamovi sa podarilo na populárnej stránke Krtkostarter™ nájsť nový druh sieťových káblov. Nové káble sú podľa ich vynálezcu, Krtka, kratšie, a teda rýchlejšie. Má to ale jeden háčik – káble fungujú iba v jednom smere. Toto ale Adam nemá problém vyriešiť a tak objednal pár zväzkov týchto káblov.

Po pár mesiacoch, troch oddialeniach prvej várky (nejaké výmysly o zamorení továrne hadmi šípkovitými – čo ešte nevymyslia) a zopár súdoch Adamovi prišiel balík s káblami. Ako to už ale býva s nakupovaním na Krtkostarter-i, nebolo to úplne to pravé orechové. Káble na sebe mali mať značenia, ktorým smerom vlastne fungujú. Po podrobnom preskúmaní si ale Adam všimol, že niektoré káble majú trochu divné označenie. Aby toho nebolo málo, niektoré káble sa začali hmýriť! Adam sa zvládne postarať o hmýriace sa hady, no musel Vás poprosiť o pomoc s triedením káblov. Pomôžte mu! (fakt si to už chce pozrieť!)

Úloha

Na každom kábli sa nachádza reťazec zložený zo šípkov – znakov '<' a '>'. Pokiaľ vynecháme všetky výskyty podreťazca susediacich “zátvorkových” šípkov (reťazec <>), ostanú nám iba šípky, ktoré indikujú správny smer kábla. Na vstupe dostanete vždy jeden kábel a Vašou úlohou bude povedať, či je kábel ľavo-dátový alebo pravo-dátový.

Formát vstupu

Na vstupe sa nachádza práve jeden riadok – reťazec znakov '<' a '>' – náš kábel. Každý vstup má zaručené, že sa skladá iba z reťazcov <> a zvyšných znakov, ktoré budú vždy iba šípky smerujúce jedným smerom, doľava ('<') alebo doprava ('>'). Adam kupoval iba kratšie káble, teda dĺžka reťazca n nepresiahne 10^6 znakov.

Formát výstupu

Na výstup vypíšete práve jeden riadok s jedným z nasledujúcich reťazcov – **ľavo-dátovy** pokiaľ je kábel ľavo-dátový, teda všetky jeho šípky, ktoré nepatria k šípkovým zátvorkám <>, mieria doľava. Nápodobne vypíšete reťazec **pravo-dátovy**, pokiaľ je kábel pravo-dátový, teda všetky jeho nezátvorkové šípky ukazujú doprava.

Hodnotenie

Je 8 sád vstupov. Platia v nich nasledujúce obmedzenia:

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

Príklad

vstup

<

výstup

ľavo-dátovy

V prvom príklade má kábel iba jednu ľavú šíпку a teda musí byť ľavo-dátový.

vstup

<><><><><>>><>>><>

výstup

pravo-datovy

Druhý prípad je už trochu zamotanejší, no môžeme si všimnúť, že po troch pároch šípok narazíme na samotnú šípku smerom doprava. Vo zvyšku reťazca narazíme na ďalšie osamelé šípky pravým smerom, no žiadnu osamelú smerom dolava, teda kábel musí byť pravo-dátový.

2. A nech sú moje hady v suchu

12 b za popis, 8 b za program

Viete čo mačky naozaj nemajú rady? Vodu. A viete aké zviera nemá rado mačky? Predsa hady. Ty, ako správny hadonoš¹, ktorý má svoje hady veľmi rád, by si chcel svojim hadom dopriať pokoj od mačiek. Po daždi sa to dá veľmi jednoducho. Na niektorých miestach Tvojho pozemku po daždi ostane stáť voda, a to sú miesta, kam môžeš dať svojich hadov a budú mať pokoj od mačiek. Teraz by si ale chcel zistiť, koľko vodou zatopených políčok ostane na tvojom pozemku po daždi.

Úloha

Pre jednoduchosť uvažujme, že Tvoj pozemok je dvojrozmerný. Má iba šírku a výšku. Každé miesto na Tvojom pozemku je súvislý stĺpec zeme od výšky 0 až po výšku terénu v danom mieste. Pôda na Tvojom pozemku je nepriepustná a vždy naprší dostatočné množstvo vody aby zaplnilo všetky miesta kde môže ostať voda. Vypíš súčet plôch tých políčok Tvojho pozemku, kde po napršaní ostane voda.

Formát vstupu

Na prvom riadku vstupu je číslo n označujúce šírku pozemku. Platí $1 \leq n \leq 50\,000$. Nasleduje jeden riadok obsahujúci n medzerou oddelených celých čísel v_i , výšky terénu v poradí zľava doprava. Platí, že $0 \leq i < n$ a $0 \leq v_i \leq 220\,000$

Formát výstupu

Vypíšte jedno číslo, počet políčok, ktoré po daždi ostanú zatopené vodou. Nezapudnite za číslom vypísať znak konca riadku.

Obmedzenia

Sú 4 sady vstupov po 2 body. Platia v nich nasledovné obmedzenia:

Sada	1	2	3	4
$1 \leq n \leq$	20	21 000	35 000	50 000
$0 \leq v_i \leq$	50	50 000	200 000	220 000

Všimnite si že vstupné a výstupné údaje sa nemusia zmestiť do 32 bitovej premennej, odporúčame preto použiť 64 bitovú premennú (napr. `long long` v C++).

Príklad

vstup

```
12
0 1 0 2 1 0 1 3 2 1 2 1
```

výstup

6

Označme X zem a 0 vodu a pre lepšiu ilustráciu – nulovú výšku terénu. V takejto krajine sa po napršaní udrží 6 vody:

```

X
X000X0X
X0XX0XXXXX
-----
```

¹Človek, ktorý nosí hady. Nemýľte si to s `hadonos`, to je nos, ktorý vyzerá ako had.

vstup

```
3
1 1 1
```

výstup

```
0
```

Tu sa nemá kde zachytiť voda a teda žiadne políčko neostane po daždi zatopené.

3. Dobré hádanie

12 b za popis, 8 b za program

Pytón Python sa rozhodol, že pri príležitosti 30. výročia vybudovania jeho štvrte usporiada veľkolepú online párty, na ktorej si všetci účastníci zahrajú novú hru s názvom Háda, Hádaš, Hádate. Princíp tejto hry spočíva v tom, že Python si pripraví množinu čísel, o ktorej nikto z účastníkov nič netuší, a súťažiacim prezradí niekoľko čísel z tejto množiny s ich pozíciami (pri zoradení od najmenšieho) v tejto množine. Vyhráva ten hádač, ktorý prvý príde na to, akým receptom Python zostrojil množinu.

Nie sme si úplne istí, ako veľmi a či vôbec budú účastníci párty z tejto Pythonovej hry nadšení, ale čo už.

Python ale potrebuje vašu pomoc. Už má vymyslený recept na zostrojenie množiny čísel, no teraz potrebuje nejaký program, do ktorého zadá pozíciu čísla v jeho množine a on mu vypíše, aké číslo sa na tejto pozícii nachádza, aby vedel súťažiacim dávať tieto indície a nemusel to sám ručne počítat.

Úloha

Pre čísla a, b, c sa Pythonova množina skladá práve z takých x , ktoré spĺňajú práve jednu z dvoch podmienok:

- a delí x , no b nedelí x .
- a, b aj c delia x .

Poznáte čísla a, b, c a m . Python sa vás postupne opýta na m pozícií v jeho množine. Pre každú z nich zistíte, aké číslo sa nachádza na danej pozícii (pri usporiadaní od najmenšieho). Pomôžete mu tak s prípravou hry a okorením výročnej párty.

Formát vstupu

Na prvom riadku vstupu sa nachádzajú 4 čísla a, b, c (parametre množiny) a m (počet otázok), pričom platí, že $1 \leq a, b, c \leq 1\,000$ a $1 \leq m \leq 100\,000$. Nasleduje m riadkov. Na i -tom z nich je číslo n_i , čiže pozícia čísla v množine, na ktorú sa Python pýta v i -tej otázke. Platí, že $1 \leq n_i \leq 10^9$.

Formát výstupu

Na i -ty riadok vypíšete odpoveď na i -tu otázku, teda n_i -te číslo množiny.

Obmedzenia

Sú 4 sady vstupov po 2 body. Platia v nich takéto obmedzenia:

Sada	1	2	3	4
$1 \leq n_i \leq$	100	10 000	10^6	10^9
$1 \leq m \leq$	100	100 000	100 000	100 000

Pozor, vstupné a výstupné údaje sa nemusia zmestiť do 32 bitovej premennej, odporúčame preto použiť 64 bitovú premennú (napr. `long long` v C++).

Príklad

vstup

```
3 6 7 3
1
2
8
```

výstup

```
3
9
42
```

Pre $a = 3, b = 6, c = 7$ vyzerá prvých 8 prvkov množiny takto: $\{3, 9, 15, 21, 27, 33, 39, 42\}$. Každé číslo je buď deliteľné všetkými tromi alebo je deliteľné 3 ale nie 6.

vstup

```
151 993 103 3
1893
2499
24
```

výstup

```
285994
377651
3624
```

4. O žabách a hadoch

10 b za popis, 10 b za program

Žabiak Michal a žabka Lucka dostali na Vianoce hru Hady a Žaby. Táto hra je pre dvoch hráčov. Jeden hráč si zvolí postupnosť štyroch farebných žiab zo šiestich možných farieb. Farby žiab sa môžu aj opakovať. Druhý hráč háda, aká je táto postupnosť. Aby to však nemal až také ťažké, po každom tipe mu prvý hráč povie dve čísla. Prvé hovorí, koľko žiab trafil presne, teda na danom mieste je žaba správnej farby. Druhé hovorí koľko žiab mu zjedli hady (teda sú na zlom mieste). Inak povedané, toto číslo hovorí o koľko viac žiab mohol trafiť, ak by svoj tip usporiadal inak. Žabiakovi Mišovi sa táto hra veľmi zapáčila, no žabku Lucku už pomaly aj prestala baviť. Žabiak si oblúbil úlohu prvého hráča, ktorý to má síce celkom jednoduché, ale za to sa môže protihráčovi smiať vždy, keď neuhádne.

Žabiak si teda naprogramoval svojho druhého hráča, s ktorým sa môže hrať, koľko len bude chcieť. Tento jeho protihráč bol však veľmi hlúpy a hádal náhodné postupnosti. Začalo to byť rýchlo nudné. Dokážete žabiakovi naprogramovať lepšieho protihráča?

Úloha

Táto úloha je interaktívna. Váš program musí vypísať svoj tip, potom si prečítať odpoveď od žabiaka a vypísať ďalší tip, a tak ďalej, až kým nebude odpoveď 4 0. Táto odpoveď znamená, že tip je správny, teda že váš program trafil všetky štyri žaby.

Formát vstupu

Po každom tipe dostanete na vstup dve medzerou oddelené čísla – počet žiab a počet hadov.

Aby testovanie fungovalo ako má, **je nutné**, aby sa po vypísaní tipu výstup presunul z pamäte na štandardný výstup pomocou príkazu `cout.flush()` v C++ alebo `sys.stdout.flush()` v Pythone. Pre iné jazyky hľadajte ekvivalent k príkazu `flush`.

Formát výstupu

Každý tip vypíšete ako 4 medzerou odelené čísla od 1 do 6, ktoré predstavujú farby štyroch žiab.

Hodnotenie

Je 10 sád vstupov. Každá sada má obmedzený počet tipov, ktoré sa môžete opýtať. Ak budete potrebovať viac tipov na zistenie riešenia, dostanete odpoveď WA. Maximálny počet otázok je v prvých sádach postupne 1000, 100, 50, 30 a 10. V ďalších sádach je počet vždy o jedno menší až po 5 v poslednej sade.

V popise riešenia sa sústreďte na to, aby bolo jasné prečo vaše riešenie vždy nájde odpoveď. Časovú zložitosť určíte v závislosti od počtu farieb a od počtu hádaných žiab.

Príklad

```
>>> 1 3 2 3
<<< 1 2
>>> 3 4 2 1
<<< 0 4
>>> 1 2 3 4
<<< 4 0
```

Heslo je 1 2 3 4.

5. Nádoby

12 b za popis, 8 b za program

Samo má doma veľa hadov. Chcel si kúpiť veľa hadíc (samíc hadov), aby neskôr mal aj veľa hádat. Tak si ich objednal z nemenovaného internetového obchodu, ale keď mu prišla zásielka, postihlo ho neprijemné

prekvapenie. Miesto hadíc (samíc hadov) mu omylom poslali hadice (hadice). No čo už, pomyslel si, nájdem si teda inú zábavu. A našiel si inú zábavu.

Úloha

Máme sústavu n nádob poprepájaných hadičkami. Na začiatku sú všetky nádoby prázdne. Nad prvou nádobou je kohútik, cez ktorý sa do tohto systému napúšťa voda. Funguje princíp spojených nádob.

Hadičkou začne voda pretekať, až jej hladina v niektorej z incidentných nádob dosiahne úroveň najvyššieho bodu danej hadičky. Vtedy sa rast hladiny v tej nádobe zastaví a začne sa napúšťať tá druhá a cez ňu prípadne ďalšie. Až sa hladiny vyrovnajú, zostanú vyrovnané navždy a ak napúšťanie pokračuje, budú stúpať súčasne.

Vašou úlohou je zistiť, z ktorej nádoby voda pretečie.

Formát vstupu

Na prvom riadku sú dve medzerou oddelené čísla n a k – počet nádob a počet hadičiek. Nádoby sú očíslované od 1 po n .

Na druhom riadku je n čísel, d_1 až d_n – výšky, v ktorých sa nachádza dno danej nádoby. Na treťom riadku je n čísel, h_1 až h_n – výšky horných okrajov jednotlivých nádob.

Nasleduje k riadkov popisujúcich jednotlivé hadičky. Na i -tom riadku je päť medzerou oddelených celých čísel, x_i, y_i, a_i, b_i, c_i . Táto hadica spája nádoby x_i a y_i . Platí $x_i \neq y_i$. Pripája sa do nádoby x_i vo výške a_i a do nádoby y_i vo výške b_i . Jej najvyšší bod je vo výške c_i . Platí $a_i < c_i > b_i$ a koncové body hadice sú vyššie ako dno nádoby ku ktorej sa pripájajú. Koncový bod však môže byť vyššie ako horný okraj nádoby, v takomto prípade cez neho môže do nádoby voda iba natekať (vytekať nie).

Medzi tou istou dvojicou nádob môže viesť aj viac ako jedna hadička.

Všetky výškové údaje sú celé čísla z intervalu od 0 do 10^9 vrátane a sú navzájom rôzne.

Formát výstupu

Vypíšte jedno číslo, číslo nádoby ktorá pretečie.

Hodnotenie

Je 8 sád vstupov. Platia v nich nasledujúce obmedzenia:

Sada	1–2	3–5	6	7–8
$1 \leq n \leq$	100	1 000	10 000	100 000
$1 \leq k \leq$	1 000	10 000	100 000	100 000

Príklad

vstup

```
3 3
20 10 0
90 50 60
1 2 25 15 35
2 3 30 40 45
1 3 70 75 80
```

výstup

```
2
```

Keď hladina v nádobe 1 dosiahne výšku 35, začne sa naplňovať nádoba 2. Potom čo aj v nádobe 2 dosiahne výšku 35, voda začne stúpať v oboch nádobách súčasne, až kým nedosiahne výšku 45, kedy sa začne napúšťať nádoba 3. Keď voda aj v nej dosiahne výšku 45, začne voda stúpať vo všetkých nádobách súčasne, až dokým nedosiahne výšku 50, keď začne pretekať cez okraj nádoby 2 a hladina už ďalej stúpať nebude nikde.

vstup

```

5 6
30 50 70 40 90
300 280 290 210 110
1 2 220 200 230
3 4 100 120 170
3 4 130 60 150
4 5 160 190 260
1 5 180 250 270
4 2 140 80 240

```

výstup

4

Hoci je nádoba 5 najnižšia, voda sa do nej nedostane, skôr sa vyleje z nádoby 4.

6. Opäť sa hady hadiť budú

12 b za popis, 8 b za program

Blíži sa jar, kvety začnú pučať, hady sa začnú hadiť. Každý čo i len trochu decentný hadonoš si teraz plánuje rozloženie svojej zbierky na hadisko. Počet možných rozložení je častokrát obrovský, avšak mnoho začínajúcich hadonošov si to vôbec neuvedomuje a príprave nevenujú dostatok času. Navyše sa potom čudujú, keď dosiahnu malý hadiaci kvocient a v lete slabú úrodu! Pre mnohých z vás je toto tiež iba prvá hadonošská jar, a povedzme si to na rovinu, vieme, že nie všetci ste z tých najzodpovednejších. Možno by ste to brali serióznejšie, keby ste presne vedeli ako *hrozne* veľa kombinácií treba zvážiť? Dokonca to možno poslúži ako dobrá rozcvička vašich mozgov pred tým, ako sa začnete trápiť s hľadaním toho optimálneho rozloženia.

Úloha

Hadisko má tvar mriežky s rozmermi $N \times M$. Vo vašej zbierke je K hadov a radi by ste ich rozložili na hadisko. Na každé políčko môžete položiť najviac jedného hada, a aby sa príliš nepohadili, do každého stĺpca a riadku nanajvýš dvoch. Pre účely hadenia môžeme všetkých hadov aj napriek ich náramne odlišným osobnostiam považovať za identických. Koľko existuje rôznych rozložení? Keďže ich môže byť ale že *naozaj veľa*, ako odpoveď uvádzajte výsledok modulo $10^9 + 7$.

Formát vstupu

Na jedinom riadku su tri celé čísla N, M, K – rozmery hadiska a počet hadov.

Formát výstupu

Na výstup vypíšte koľko existuje rôznych rozložení 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
$1 \leq N, M \leq$	5	20	50	100
$0 \leq K \leq$	5	10	100	200

Príklady

vstup

1 1 1

výstup

1

vstup

2 2 2

výstup

6

vstup

3 3 3

výstup

78

vstup

4 4 4

výstup

1428

7. Šibalské sny

12 b za popis, 8 b za program

Súhvezdia sú nudné. Existujú si na oblohe a veľa toho nenarozprávajú. Niektoré dvojice hviezd v súhvezdí sú spojené čiarou, vraj, aby ľuďom viac niečo pripomínali.

Sem-tam sa ale aj také súhvezdie potrebuje ponatahovať. Hadonoš² rád sníva. Rozmýšľa, ako rôzne by sa mohol ponatahovať, keby bol nejakým iným súhvezdím. Vždy, keď sa mu takto prisní, že je iným súhvezdím, prudko sa zobudí a začne počítať, koľkými rôznymi spôsobmi by vedel svoje hviezdy usporiadať. Nemá ale príliš veľký mozog a tak sa mu snívajú iba samé spojité acyklické konfigurácie.

Nie je to ale iba tak. Nebyť súhvezdí, Kolumbus by nedoplával do Ameriky⁴, Cook by zablúdil už pri Dubline a Krtko by nevedel trafiť na správne sústredenie. Nemôžu sa teda natahovať len tak, ako sa im zachce. Musia sa ľuďom javiť nezmenené.

Úloha

Hadonoš vám postupne popíše každý jeho sen. Každý sen je jedno súhvezdie, teda hviezdy a čiary, ktorými sú niektoré dvojice hviezd prepojené. Každé prisnené súhvezdie tvorí strom. Každá hviezda má svoje číslo, hviezdy sú teda rozoznateľné.

Hadonoš po zobudení potrebuje vyrátať, koľkými spôsobmi sa dajú v prisnenom súhvezdí preusporiadať hviezdy tak, aby platilo:

- Hviezda číslo 0 nezmení svoju pozíciu
- Množina pozícií na oblohe, ktoré boli obsadené hviezdami súhvezdia je pred usporiadaním rovnaká ako po ňom
- Ak pred usporiadaním boli čiarou spojené hviezdy a a b , tak sú spojené čiarou aj po usporiadaní

Každé súhvezdie teda tvorí strom zakorenený v 0.

Bez odpovede Hadonoš znovu nezaspí a na zajtrajšiu skúšku z astronómie príde unavený. Uspite ho!

Formát vstupu

Na prvom riadku vstupu sa nachádza číslo T – počet Hadonošových snov. Platí $1 \leq T \leq 10$. Nasleduje T popisov snov.

Popis sna začína riadkom s číslom n – počet hviezd v prisnenom súhvezdí. Platí $3 \leq n \leq 10\,000$. Nasleduje $n - 1$ riadkov. Na i -tom z nich sa nachádzajú čísla a_i, b_i oddelené medzerou. Tie hovoria, že v prisnenom súhvezdí sú čiarou spojené hviezdy a_i a b_i . Platí $0 \leq a_i, b_i < n$.

Každé prisnené súhvezdie tvorí strom.

Formát výstupu

Pre i -ty sen vypíšte na i -ty riadok výstupu jedno číslo – počet rôznych usporiadaní hviezd i -teho prisneného súhvezdia spĺňajúcich vyššie popísané požiadavky. Keďže odpoveď môže byť veľmi veľká, vypisujte ju modulo $10^9 + 7$.

Hodnotenie

Sú 4 sady vstupov. Platia v nich nasledovné obmedzenia:

Sada	1	2	3	4
$3 \leq n \leq$	8	20	1 000	10,000

²Nemýliť s hadonos³.

⁴Možno...

Príklad

vstup

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

výstup

```
2
8
```

V prvom súhvezdí sú vyhovujúce dve možnosti. Prvou je nespraviť žiadnu zmenu, druhou je vymeniť vrcholy 3 a 4. V druhom súhvezdí je jednou z možností napríklad vymeniť koreňu ľavý a pravý podstrom a navyše vymeniť vrcholy 5 a 6.

8. Inovatívny dažď

12 b za popis, 8 b za program

Hady sužované neustálymi dažďami dali jedného dňa hlavy dohromady a vynašli Hadí Automat Aktualizujúci Rozbité Počasie. Problém je, že je to iba jednoduchý prototyp, ktorý dokáže jednu vec: vyjasní počasie na nejakom obdĺžnikovom kuse zeme. Následne sa hady znovu zamysleli a dokázali upraviť Hadí Automat Aktualizujúci Rozbité Počasie, aby slnil vo viac *neprekrývajúcich* sa obdĺžnikoch naraz.

Hady by chceli použiť Automat na oslňovanie parku. Park si vieme predstaviť ako štvorcovú plochu s n lavičkami a $n - 1$ cestičkami spájajúcimi lavičky. Navyše, cestičky sú postavené tak, že z každej lavičky sa vieme po cestičkách dostať ku ktorejkoľvek inej lavičke (teda cestičky tvoria strom).

Hady väčšinou využívajú park tak, že sa rozležia po nejakej ceste: položia hlavu na jednu lavičku, chvost na druhú lavičku (alebo to môže byť had veľmi krátky a ležiaci len na jednej lavičke) a zvyšok teda na lavičky cestou medzi týmito dvoma lavičkami. Predtým nastaví Automat, aby oslnil niekoľko obdĺžnikov tak, aby oslnil práve tie lavičky kde had leží, ale nie tie prázdne.

Ako sa had ukladá na slnenie, s hrôzou si uvedomí, že si nepamätá, či nastavil Automat správne a bude slniť práve tie lavičky kde had leží. Pomôžte mu!

Úloha

Existuje n lavičiek pospájaných $n - 1$ cestičkami. Lavičky ležia na štvorcovej ploche rozmerov $n \times n$.

Postupne prichádza q hadov. Pre každého hada poznáme lavičku, na ktorej má hlavu, lavičku, na ktorej má chvost a konfiguráciu Automatu. Konfigurácia Automatu pozostáva z niekoľko neprekrývajúcich sa obdĺžnikov, ktoré majú byť slnené. Pre každého hada zistíte, či Automat nakonfiguroval správne, teda či obdĺžniky obsahujú práve tie lavičky, na ktorých leží had.

Hady využívajú park v dostatočných odstupoch po sebe, takže sa nemusíte trápiť, či sa hady neprekrývajú alebo neslnia iného hada. Žiadne predchádzajúce nastavenia Automatu neovplyvňujú nadchádzajúce slnenia.

Formát vstupu a výstupu

Na prvom riadku vstupu je číslo n – počet lavičiek, $1 \leq n \leq 40\,000$.

Na ďalších n riadkoch sú súradnice lavičiek. Na i -tom z nich sú medzerami oddelené celé čísla x_i a y_i – súradnice lavičky, $1 \leq x_i, y_i \leq n$.

Na ďalších $n - 1$ riadkoch je popis cestičiek: pre každú cestičku na samostatnom riadku sú dve čísla oddelené medzerou, a a b , $1 \leq a, b \leq n$, $a \neq b$.

Ďalší riadok obsahuje číslo q – počet slniacich sa hadov, $1 \leq q \leq 100\,000$.

Pre každého slniaceho sa hada dostanete najskôr tri čísla oddelené medzerou a , b a k . $1 \leq a, b \leq n$ – postupne čísla lavičky kde má had hlavu a lavičky, kde má chvost, a $1 \leq k \leq 3$ – počet slnených obdĺžnikov. Nasledujúcich k riadkov obsahuje popis obdĺžnikov: štyri medzerou oddelené čísla x_1 , y_1 , x_2 a y_2 – súradnice najskôr ľavého dolného a potom pravého horného bodu obdĺžnika. $x_1 \leq x_2$ a $y_1 \leq y_2$. Platí že bod (x, y) je v tomto obdĺžniku práve ak $x_1 \leq x \leq x_2$ a $y_1 \leq y \leq y_2$. Je garantované, že obdĺžniky sa neprekrývajú. To znamená, že neexistuje žiadny bod (x, y) , ktorý je v dvoch rôznych obdĺžnikoch.

Pre každého hada vypíšte jeden riadok obsahujúci OK, ak je Automat správne nakonfigurovaný, a NIE ak niečo je zle.

Hodnotenie

Je 8 sád vstupov.

- V prvých troch sádach platí $1 \leq n, q \leq 1000$
- V prvej, štvrtej a piatej sade navyše platí, že cestičky vždy spájajú lavičky s číslami i a $i + 1$
- V piatej a šiestej sade navyše platí, že $k = 1$ pre každého hada
- V siedmej a ôsmej sade neplatia žiadne ďalšie obmedzenia

Neodporúčame pokúšať sa úlohu riešiť v pomalších jazykoch, ako napríklad Python.

Príklad

vstup

```
9
1 1 // Pozície lavičiek
1 5
3 4
4 1
4 2
5 1
1 6
2 6
1 7
1 2 // Cesticky
1 3
1 4
2 7
4 5
4 6
7 8
7 9
5
1 9 1
1 1 2 7
1 9 1
1 1 1 7
8 3 2
1 3 4 6
1 1 2 2
5 8 3
1 1 4 2
1 3 2 6
5 5 8 9
3 5 1
3 1 4 4
```

výstup

```
NIE
OK
OK
OK
NIE
```

V prvom prípade jediný obdĺžnik slní lavičku číslo 8, čo nechceme. V druhom prípade je rovnaký had, ale zmenšený slnený obdĺžnik. Všimnite si v treťom prípade, že obdĺžnik nemusí pokrývať súvislú časť hada (táto otázka by sa nemohla objaviť v sade 7). V štvrtej otázke máme tretí obdĺžnik úplne zbytočne, – neslíní žiadne lavičky. Piaty had zabudol oslniť lavičku 1.

Znaky // označujú komentáre, ktoré sa v skutočnom vstupe neobjavia. Slúžia len na objasnenie vstupu.