



Korešpondenčný seminár z programovania

Leták letnej časti XLIII. 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 mesiac a pol 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 neprezrádzaj riešenia ostatným riešiteľom. Odpisovanie riešení a prezradenie riešení pred termínom kola je porušením pravidiel KSP. Po skončení kola sa, samozrejme, o riešeníach 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 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**. 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 sústreďenie.

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 letnej časti

Termín odoslania riešení tohto kola je **16. júna 2025**. Doprogramovávanie končí 30. júna 2025.

1. Podte si adoptovať psíkov

12 b za popis, 8 b za program

V miestnom útulku funguje adopcia psíkov¹ trochu zvláštne. Ľudia aj psíkovia sa postavia do radu. Vždy keď susedí človek s nejakým psíkom, tak sa môže rozhodnúť, že si ho adoptuje. Toto sa opakuje, pokiaľ sa to dá. Majiteľa útulku by ale zaujímalo, koľko najmenej smutných psíkov a ľudí vie ostať na konci dňa. Viete mu s tým pomôcť?

Úloha

Na vstupe dostanete postupnosť znakov P a C. Vždy, keď sa v postupnosti vedľa seba nachádza P a C, tak ich môžeme vymazať. Toto opakujeme pokiaľ to ide. Koľko najmenej znakov nám vie ostať na konci?

Formát vstupu

Na prvom riadku je celé číslo N ($1 \leq N \leq 10^6$). Na druhom riadku je reťazec dĺžky N pozostávajúci z písmen P a C.

Sada	1	2	3	4
$1 \leq N \leq$	100	10'000	10^6	10^6

Formát výstupu

Vypíšte jeden riadok a v ňom jedno celé číslo, ktoré je minimálny počet znakov, ktoré vedia ostať na konci.

Príklady

vstup

2
PC

Prvý človek si adoptuje prvého psa.

vstup

4
CCPP

Najprv si adoptuje vnútorný človek vnútorného psa a následne vonkajší človek bude susediť s vonkajším psom, čiže si ho môže adoptovať.

vstup

3
PCP

Jednému človeku sa bohužiaľ neujde žiaden psík na adoptovanie.

výstup

0

výstup

0

výstup

1

2. Sssssss, tak robia hady

12 b za popis, 8 b za program

Martin sa veľmi dobre pozná s ostatnými gréckymi bohmi. Raz sa však rozprával so svojím starým známym HADesom, ktorý počas celej noci strážil brány Olympu. Po dlhej šichte z neho vyliezlo iba:

Sssssssssss SSSssssS SSssssssssSS SSSSSSS SSSSSsSSs?! sssssssSS ssSSSSSSSSsSS SSSSSSSs sssssssss sssSSSSSSS SSSSSSS SSSSSssS. SSSS SSSSSssss SSSssssSSS SSSssSSSSs sSSssss?

¹V prípade záujmu o adopciu alebo podporu sa môžete obrátiť na [Slobodu zvierat](#)² alebo na Váš miestny útulok.

Určite každý z vás ovláda hadie jazyky, Pythony a iné Kobry, ale pre tých z vás, ktorí zadaniu vyššie nerozumejú, je tu aj jeho preklad do reči smrteľníkov.

Zadanie:

Vašou úlohou je odpovedať na základe pohybu hadíka v mriežke s rozmermi $m \times n$, na koľkých políčkach mohol začať bez toho, aby vyšiel mimo mriežku. Hadík sa každým kolom zväčšuje a pohybuje sa do jedného zo štyroch základných smerov. Ak je postupnosť na vstupe neplatná (hadík narazil sám do seba alebo vyšiel mimo mriežku), odpovedzte 0.

Formát vstupu

V prvom riadku vstupu dostanete dve čísla m - počet stĺpcov, n - počet riadkov, Nasleduje 1 riadok v ktorom dostanete reťazec dĺžky t obsahujúci znaky: 'H', 'D', 'P', 'L' - reprezentujúce smery: hore, dole, vpravo, vľavo.

V sadách platia nasledujúce obmedzenia:

Sada	1	2	3	4
$1 \leq m, n \leq$	20	10^3	10^3	10^9
$1 \leq t \leq$	20	10^3	10^5	10^5

V sadách 1, 2 navyše platí že hadík sa pohybuje iba dole a doprava.

Formát výstupu

Na jeden riadok vypíšete jedno číslo - na koľkých políčkach mohol had začať. *Keďže výsledok môže byť fakt veľký nezabudnite použiť `long long` v C++*

Príklady

vstup	výstup
<pre>3 3 DDD</pre>	<pre>0</pre>
<i>Nezáleží, kde hadík začne v mriežke, vždy zide o 3 políčka nižšie (mimo mriežku)</i>	
vstup	výstup
<pre>4 3 DDPPHLL</pre>	<pre>0</pre>
<i>Hadík narazil sám do seba</i>	
vstup	výstup
<pre>13 13 PDDDPDDDDPPPPPPPPDDDD</pre>	<pre>9</pre>

3. Ľš zrozumiteľne?

12 b za popis, 8 b za program

Na Záhorí kdesi žije fena Marta (korelácia s reálnymi rozprávkami je čisto náhodná). Jej páničkovia ju krmia záhadnou písmenkovou polievkou. Keď Marta zje túto záhadnú polievku špecifickým spôsobom, začne rozprávať. Všetci si nad tým lámali hlavu. Po chvíli zistili, že Marta rozpráva iba vtedy keď zjedla celú polievku – ale s jedným háčikom. Musela ju jesť nasledovne: každé prehltnutie začína a končí rovnakým písmenkom. Marta je ale rýchly jedlík, tak polievku chce zjesť čo najrýchlejšie (za čo najmenej hltov). Na ďalší deň sa Marta zobudila, ale už nevedela rozprávať. Tak si šla opäť dať polievku, ale zistila, že páničkovia dnes navarili inú. Poradte jej, ako ju najefektívnejšie zjesť tak, aby znova rozprávala.

Úloha

Máme reťazec dĺžky n .

Tento reťazec musíme postupne, krok po kroku, vymazať. Mazanie funguje nasledovne:

V jednom kroku si vyberieme súvislú podpostupnosť reťazca začínajúcu a končiacu rovnakým písmenkom a zmažeme ju. Takto dokážeme vymazať aj samostatné písmenko, keďže začína a končí rovnakým písmenkom.

Nájdite minimálny počet krokov, za ktoré môžeme vymazať reťazec.

Formát vstupu

Na jedinom riadku vstupu sa nachádza reťazec s dĺžky n .
 s obsahuje iba veľké a malé písmenká anglickej abecedy.

Formát výstupu

Na výstup vypíšete jediné číslo - minimálny počet krokov, za ktoré môžeme vymazať reťazec.

Obmedzenia

Vstupy úlohy sú tvorené štyrmi sadami s nasledujúcimi obmedzeniami:

Sada	1	2	3	4
$1 \leq n \leq$	$5 \cdot 10^5$	50	1000	$5 \cdot 10^5$

V prvej sade navyše platí, že s obsahuje iba písmenká **a** a **b**.

Príklady

vstup	výstup
abba	1
<i>1. vymažeme 'a..a' -> reťazec po vymazaní : ''</i>	
vstup	výstup
abcbaccf	3
<i>1. vymažeme 'abcba' -> reťazec po vymazaní : 'ccf'</i>	
<i>2. vymažeme 'cc' -> reťazec po vymazaní : 'f'</i>	
<i>3. vymažeme 'f' -> reťazec po vymazaní : ''</i>	
vstup	výstup
abacaddc	2
<i>1. vymažeme 'aba' -> reťazec po vymazaní : 'caddc'</i>	
<i>2. vymažeme 'caddc' -> reťazec po vymazaní : ''</i>	

4. Koľko strihaní potrebujeme?

12 b za popis, 8 b za program

Pokojne sedíte vo svojej kancelárii a vychutnávate si kávu. Ste radi, že vás práve povýšili na CEO celoštátnej firmy „Kolosálne Spojenie Psikov“ a užívate si výhľad zo svojho okna. Spomínate si, ako ste začínali pracovať pre túto firmu, ktorá rozdeľuje elektrinu do všetkých miest nášho milovaného štátu. Pamätáte si výstavbu masívnej elektrárne a pokladanie káblov medzi našimi malebnými mestami.

Zrazu začujete krik a do vašej kancelárie vbehne vaša sekretárka Miška s desivou správou: gigantická elektrárneň má problémy a jej produkcia elektriny kolíše. To môže viesť k preťaženiu siete a následným explóziám rozvodných staníc. Vypnúť elektrárneň nie je možné, pretože by to zruinovalo vašu firmu. Našťastie vás napadne riešenie:

Každé mesto má rozvodnú stanicu, ktorú viete na diaľku vypnúť. Tým odpojíte mesto od siete, čím riskujete kritiku. Aby ste predišli kolapsu, potrebujete vypínať mestá v takom poradí, že ak odpojíte od siete prvých j miest v poradí, všetky zvyšné mesta zostanú spojené s elektrárňou.

Úloha

Elektrická sieť pozostáva z n miest, očíslovaných od 0 po $n - 1$, m káblov a práve jednej elektrárne, ktorá sa nachádza v meste k . Elektrina prúdi od elektrárne do všetkých miest, do ktorých sa z nej dá dostať káblom (káble vedú prenášať energiu obojsmerne).

Keď vypneme mesto od elektriny, vypneme všetky káble ktoré cez neho vedú, a teda cez neho nesmie ani prúdiť elektrina do iných miest.

Nájdite postupnosť v ktorej vieme odpájať mestá od elektriny, tak aby pre každé $1 \leq j \leq n$ platilo, že ak od elektriny odpojíme prvých j miest, neovplyvníme tak elektrinu do zvyšných $n - j$ miest.

Všimnite si, že ak od elektriny odpojíte mesto s elektrárňou, do ďalších miest už elektrina ísť nevie (nemá odkiaľ).

Formát vstupu

- Prvý riadok obsahuje čísla $1 \leq n \leq 10^5$ (počet miest), $0 \leq m \leq \min(2 \cdot 10^5, \frac{n(n-1)}{2})$ (počet káblov) a $0 \leq k \leq n - 1$ (číslo mesta s elektrárnou), oddelené medzerou.
- Nasleduje m riadkov s dvojicami čísel $0 \leq a_i \neq b_i \leq n - 1$ oddelených medzerou, označujúcich, že medzi mestami a_i a b_i sa nachádza kábel.

Je garantované, že v zadanej elektrickej sieti má každé mesto prístup k elektrine.

Formát výstupu

Vypíšte jeden riadok v ktorom sa nachádza n medzerami oddelených čísel: postupnosť v ktorom vieme odpájať mestá od elektriny.

V prípade, že existuje viac postupností, vypíšte ľubovoľnú.

Malé varovanie

V prípade, že používate python a rekurziu, nezabudnite, že prednastavený limit rekurzie v pythone je 1000, a pre väčšie hĺbky ju treba manuálne nastaviť v programe cez `sys.setrecursionlimit`.

Bodovanie a obmedzenia

Existujú štyri testovacie sady, každá za dva body, v ktorých platia nasledovné obmedzenia:

Sada	1	2	3	4
$1 \leq N \leq$	10	10^5	200	10^5
$1 \leq M \leq$	45	10^5	1 000	$2 \cdot 10^5$

Zároveň, v druhej sade navyše platí, že každé mesto je prepojené káblami s práve dvoma inými mestami.

Príklady

vstup

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

výstup

```
0 2 1
```

Tento vstup by sa mohol nachádzať v druhej sade. Keď odpojíme od energie mesto číslo 0, elektrina vie medzi elektrárnou (v meste 1) a mestom 2 prúdiť cez priamy kábel. Keď následne odpojíme mesto 2, v meste 1 stále je elektrina, lebo sa v ňom nachádza elektrárňa. Napokon, keď vypneme elektrárňa, ovplyvní to len to mesto, v ktorom sa nachádza.

vstup

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

výstup

```
0 1 2
```

Všimnite si, že nemôžeme najskôr odpojiť mesto 1, keďže jediný spôsob, akým sa elektrina vie dostať do mesta 0 je cez mesto 1.

vstup

```
2 1 0
0 1
```

výstup

```
1 0
```

Nemáme na výber. Najskôr musíme vypnúť elektrinu v meste bez elektrárne

vstup

```
1 0 0
```

výstup

```
0
```

Máme jediné mesto, v ktorom je elektrárňa.

vstup

```

8 11 2
1 0
2 0
1 6
7 1
6 7
2 6
3 2
4 5
5 2
3 4
2 1

```

výstup

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

Je možné že pre nejaký vstup existuje viacero správnych riešení. Akékoľvek z nich bude akceptované.

5. Obrovitánsky smäd

12 b za popis, 8 b za program

Fipo raz pozeral [video](#)³ o tom, ako sa odlišuje spôsob pitia u psov a mačiek. Uvedomil si pritom, že napriek tomu, že mnohým z nás sa aj tak pri pití podarí nechtiac sa obliať, to v porovnaní so zvieratami nemáme až také ťažké. O koľko jednoduchšie by to však bolo, keby namiesto toho voda tiekla smerom ku nim!

Zrazu si predstavil horu, na ktorej bol prameň, z ktorého cez mnohé údolia tiekol potok smerom nadol. V niektorých údoliach sa ale nachádzali psíky pachtiace za každou kvapkou z prameňa, ktorá k nim stiekla. Nanešťastie bol však prameň už takmer vyschnutý a ešte k tomu sa jeho tok stále menil a prechádzal z údolia do údolia...

Úloha

Hora, ktorú si Fipo predstavil, mala n údolí očíslovaných od 0 po $n - 1$. V údolí 0 zároveň pramení potok. Z prameňa postupne po jednej steká ďalej t kvapiek. Kvapky tečú údoliami a vždy, keď prídu do nejakého údolia, kde nie je psík, stečú do nejakého iného údolia pod ním. Ak je v nejakom údolí psík, tento psík kvapku vypije.

POZOR: Môže existovať aj vyššie položené údolie ako prameň (teda také, z ktorého by kvapky stiekli do údolia s prameňom).

Pre každé údolie i poznáte zároveň hodnotu k_i . Ak $k_i = 0$, tak je v údolí psík. Inak dostanete pre údolie i aj postupnosť údolí A_i . Po každých k_i kvapkách, ktoré pritečú do údolia i , sa zmení to, do ktorého údolia kvapky odtiaľto potečú. Je to nasledujúce údolie v postupnosti A_i , pričom prvých k_i kvapiek potečie do údolia $A_i[0]$, druhých k_i kvapiek do $A_i[1]$, a tak ďalej... Zároveň viete, že postupnosť A_i je dlhá $\lceil t/k_i \rceil$ (teda zaokrúhlené nahor). Všimnite si, že pri takejto dĺžke postupnosti A_i sa nikdy nestane, že by niektorá z uvažovaných t kvapiek už nemala kam ďalej tiecť.

Hoci sú psíky veľmi smädné, fyziku aj napriek tomu ešte zatiaľ nevedia oklamať. Preto voda v každom okamihu steká len z vyššie položeného údolia do nižšie položeného, takže nevie tiecť do cyklu. Zistite, koľko kvapiek sa napokon dostane ku každému psíkovi.

Formát vstupu

V prvom riadku vstupu dostanete medzerou oddelené čísla n a t - počet dolín a počet kvapiek stekajúcich z prameňa.

V každom z nasledujúcich n riadkov je celé číslo k_i ($0 \leq k_i \leq 2 \cdot 10^9$). Ak $k_i = 0$, znamená to, že sa v danom údolí nachádza psík. Inak hodnota k_i určuje, po koľkých kvapkách začne voda stekať do ďalšieho údolia. Na zvyšku riadku je $\lceil t/k_i \rceil$ čísel, pričom m -té z nich určuje, do ktorého údolia bude stekať $m \cdot k_i$ -ta až $(m + 1) \cdot k_i - 1$ -vá kvapka, ktorá do tohto údolia pritečie (m číslujeme od nuly).

Nech z_i je počet údolí, do ktorých niekedy bude tiecť voda z údolia i . Označme potom $\sum z_i$ súčet hodnôt z_i cez všetky údolia i , v ktorých sa nenachádza psík. Sú 4 sady vstupov, pričom v jednotlivých sadách platia nasledujúce obmedzenia:

Sada	1	2	3	4
$1 \leq n \leq$	1 000	1 000	$3 \cdot 10^4$	10^5
$0 \leq \sum z_i \leq$	3 000	3 000	10^5	$3 \cdot 10^5$

³<https://www.youtube.com/watch?v=30wkK8B4TjA>

Sada	1	2	3	4
$1 \leq t \leq$	1 000	10^9	10^9	10^9

Formát výstupu

Pre každého psíka na samostatný riadok vypíšte, koľko kvapiek sa k nemu dostane. Tieto výsledky vypíšte v rovnakom poradí, v akom sú na vstupe údolia so psíkmi.

Príklady

vstup	výstup
<pre>4 10 3 1 2 3 2 0 2 3 1 3 1 3 0</pre>	<pre>5 5</pre>

Prvé tri kvapky pôjdu k prvému psíkovi. Ďalšie tri kvapky potom pôjdu najprv do údolia 2 a odtiaľ prvé dve k druhému psíkovi a tretia k prvému psíkovi. Ďalšia trojica kvapiek pôjde priamo k druhému psíkovi. Posledná kvapka pôjde najprv do údolia 2 a odtiaľ k prvému psíkovi, keďže je to už štvrtá kvapka, ktorá sa dostala do údolia 2.

vstup	výstup
<pre>3 6 9 2 0 0</pre>	<pre>0 6</pre>

Z prameňa sa spustí 6 kvapiek, no až po 9 kvapkách sa má zmeniť smer odtekania vody, preto všetky kvapky skončia pri druhom psíkovi.

vstup	výstup
<pre>3 17 0 21 0 0</pre>	<pre>17 0</pre>

Priamo v údolí s prameňom sa nachádza psík, preto všetky kvapky vypije tento psík. Všimnite si, že údolie s prameňom sa nenachádza v najvyššie položenom údolí, keďže by doň stekala voda z údolia 1, ak by sa do údolia 1 nejaké kvapky dostali.

6. Vyrovnaná fotka

12 b za popis, 8 b za program

Jedného dňa prebehlo rokovanie rady starších Konglomerátu Sebececkých Podnikateľov. Prebralo sa veľa vecí. Na nejakých sa zhodnúť vedeli, na iných nevedeli. Zhodli sa však na jednom. Potrebujú spoločnú fotku zamestnancov konglomerátu (nie nutne všetkých). Samozrejme, má to byť reprezentatívna fotka. Konglomerát Sebececkých Podnikateľov má na ňu nemalé požiadavky tiež približné predstavy, ako by mala vyzerat. Bolo vybraných niekoľko reprezentatívnych zamestnancov, ktorí by mali byť na fotke, a tiež ich poradie. Teraz je ale medzi nejakými vedľa seba stojacimi zamestnancami veľmi veľký výškový rozdiel, a to by páni byrokrati radi opravili. Ako to už býva, nič nie je zadarmo.

Za nejakú sumu vie konglomerát zaplatiť ďalšiemu zamestnancovi, a zamestnanec príde do radu na fotku na miesto, ktoré mu vyberú. V Konglomeráte je veľa zamestnancov všetkých možných rozmerov, teda je možné zohnať zamestnanca akejkoľvek výšky. Tiež je možné za určitú cenu poslať preč už vybraného zamestnanca, ktorého chcela mať rada starších na fotke, z radu na fotku preč. Za nejakú, stále trochu inú cenu, je dokonca možné poslať nejakého z už vybraných zamestnancov na operáciu v najnovšej nemocnici Konglomerátu Sebececkých Podnikateľov, ktorá vie zmeniť výšku zamestnanca.

Pomôžte Konglomerátu Sebececkých Podnikateľov a zistite, ako najlacnejšie vedľa rad na fotku upraviť tak, aby nebol medzi žiadnymi vedľa seba stojacimi zamestnancami príliš veľký výškový rozdiel.

Úloha

Je zadaná postupnosť čísel A_1, A_2, \dots, A_n dlhá n prvkov a čísla M, I, D . Zadané sú aj povolené úpravy, z ktorých každá niečo stojí. Vašou úlohou je za čo najmenšiu cenu pomocou nich upraviť danú postupnosť tak,

aby sa žiadne susediace prvky v upravenej postupnosti nelíšili o viac ako M . Povolené úpravy a ich ceny sú nasledovné:

- Vloženie ľubovoľného celého čísla na ľubovoľné miesto v aktuálnej postupnosti za cenu I .
- Odstránenie ľubovoľného prvku aktuálnej postupnosti A_i za cenu D .
- Zmena ľubovoľného prvku aktuálnej postupnosti A_i na x za cenu $|A_i - x|$.

Vypíšte minimálnu cenu, za ktorú vieme postupnosť upraviť do požadovaného tvaru.

Formát vstupu

V prvom vstupe riadku sú celé čísla n, M, I, D ($1 \leq n \leq 50, 0 \leq M, I, D \leq 10^9$). V druhom riadku je medzerami oddelená postupnosť A_1, A_2, \dots, A_n ($0 \leq A_i \leq 50000$).

V jednotlivých sadách platia nasledujúce obmedzenia:

Sada	1	2	3	4
$1 \leq n \leq$	5	50	50	50
$1 \leq \max A_i \leq$	10	50	300	50000

V prvej sade navyše platí, že $\max A_i \leq D$.

Formát výstupu

Vypíšte jeden riadok a v ňom jedno celé číslo - najmenšia cena, za ktorú vieme postupnosť upraviť do požadovaného tvaru.

Príklady

vstup

```
4 2 1 10
1 8 3 9
```

výstup

```
6
```

Môžeme najprv vložiť čísla 3 a 5 medzi 1 8, znížiť 8 na 7, zväčšiť 3 na 5 a vložiť 7 medzi posledné dva prvky aktuálnej postupnosti (teraz 5, 9). Takto dostaneme postupnosť 1 3 5 7 5 7 9, kde sa každé 2 susedné prvky líšia najviac o 2. Za úpravy sme zaplatili 6. Tento vstup sa môže objaviť aj v prvej sade, keďže $\max A_i \leq D, n \leq 5$ a $\max A_i \leq 10$

vstup

```
3 2 1 2
1 10 5
```

výstup

```
3
```

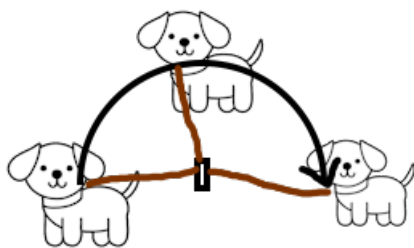
7. Ideálny Žbirkov život

12 b za popis, 8 b za program

Ako každý vie, Meko Žbirka miluje psíkov. Väčšinou ich necháva voľne vypustených vo svojej záhradke v Slovakistane, ale kvôli veľkému nárastu kriminality a krádeží v krajine sa uchýlil k bezpečnejšej alternatíve. Každého psa Meko priviazal k jednej z dvoch gigantických, kompletne čiernych železných tyčí, ktoré mu trčali v záhrade.

Psom sa tento nápad zo začiatku nepáčil. Časom si na to však až tak zvykli, že sa spolu úplne prestali stýkať psíkovia, ktorí boli priviazaní k rôznym tyčiam. Dokonca si podľa toho aj zmenili farbu srsti – pri jednej tyči na bielu a pri druhej na čiernu.

Po dlhšej dobe však psom omrzelo nadalej prechádzať okolo tyče, aby sa stretli so svojimi kamarátmi. Vymysleli si teda nový spôsob navštevovania. Pes len strašne silno vyskočí, a keďže je k tyči priviazaný, spraví veľký polkruh vzduchom a dopadne presne na opačné miesto na druhej strane tyče. Asi takto:



Psíkovia sú veľmi spoločenský, preto boli od začiatku prezieravo rozostavení tak, aby každý z nich mohol týmto pohybom stretnúť na opačnej strane tyče nejakého psíka rovnakej farby.

Teraz dostal Meko úžasný nápad. Odfotil svojich psíkov v záhradke svojím novým čierno-čiernym fotoaparátom. Nedošlo mu však, že na takejto fotke sa stratí farba psov a dokonca aj pozícia kompletne čiernych tyčí. Teraz by sa však chcel pochváliť ostatným legendárnym spevákom a ukázať im jeho psov na fotke v plnej kráse aj s tyčami, ku ktorým sú priviazaní. Keďže Meko je spevák a nebude sa zaneprázdňovať takými trápnyimi úlohami, prischla úloha vám.

Úloha

Psíkov vieme interpretovať ako body v rovine. Z Mekofo fotky vieme zistiť pozície jednotlivých psov, avšak nie pozíciu tyčí, ku ktorým sú priviazaní. Vieme, že pes je buď čierny alebo biely a všetci psíkovia jednej farby sú spolu priviazaní k jednej tyči, ktorá je taktiež bodom v rovine. Zároveň vieme, že pre každého psa existuje na opačnej strane tyče (tej, ku ktorej je priviazaný) pes rovnakej farby. Tyč sa môže nachádzať na pozícii psa. V takom prípade, ak je k nej daný pes priviazaný, je sám sebe na opačnej strane tyče.

Pre zadané pozície psov určite, či existujú také dve tyče a rozdelenie psov do farieb, ktoré spĺňajú podmienky uvedené vyššie.

Formát vstupu

V prvom riadku vstupu je číslo n ($1 \leq n \leq 300$) udávajúce počet psov na fotke.

Na i -tom z nasledujúcich n riadkov sú 2 celé čísla x_i, y_i ($|x_i|, |y_i| \leq 10^9$) - x a y súradnica i -teho psa. Žiadna z dvojíc sa na vstupe nevyskytne viac krát - psy sú na rôznych miestach.

Formát výstupu

V prípade, že neexistujú dve tyče a rozdelenia psov do farieb, ktoré spĺňajú podmienky zadania, vypíšte na jediný riadok výstupu **zbirka dozbirkoval**.

V opačnom prípade vypíšte 3 riadky.

- Na prvom riadku **zbirka je kral**.
- Na druhom riadku štyri čísla oddelené medzerami $tx_0 ty_0 tx_1 ty_1$ - súradnice prvej a druhej tyče v poradí (x a y).
- Na treťom riadku pre každého psa vypíšte bez medzier jedno číslo, buď 0, ak má byť priviazaný k tyči 0 alebo 1 ak má byť priviazaný k tyči 1.

Ak existuje viac riešení, môžete vypísať ľubovoľné z nich. Súradnice tyčí vypíšte s dostatočnou presnosťou za desatinnou čiarkou.

Hodnotenie

Sada	1	2	3	4	5	6
$1 \leq n \leq$	15	30	70	70	300	300
$0 \leq \ x_i\ , \ y_i\ \leq$	100	100	1000	10^9	10^9	10^9
dodatočné obmedzenia $x_i = y_i$						

Príklady

vstup

```
6
1 1
2 2
4 4
5 5
10 -5
10 -100
```

výstup

```
zbirka je kral
3 3 10 -52.5
000011
```

vstup

```
5
1 1
5 5
10 -5
10 -100
100000 10000
```

výstup

```
zbirka dozbirdoval
```

Jedna tyč musí byť medzi 1. a 2. psom, jedna medzi 3. a 4. a pre 5. už nič nezostane.

8. Absolútne Pestákovanie

12 b za popis, 8 b za program

Medvied a Dan majú strašne radi psíkov, a sú aj nadšený chovatelia. Po všetkých tých rokoch im postupne pribúdali psíky, až ich pomaly nemali kam dávať, lebo im postupne dochádzalo miesto na ich záhradke, keďže novým psíkom vždy treba postaviť na záhradke búdku.

A dnes prišiel ten obávaný deň! Medvied totiž domov doniesol jedno malé chutnučké šteniatko, ale s Danom zistili, že ho nemajú kam ubytovať! Treba teda rozvrhnúť nový plán bývania.

Po dlhej diskusii sa dohodli, že psíky môžu bývať v novom poschodovom dome, ktorý im teda s Danom postaví niekde na záhrade. Majú dvoch kandidátov - v rohu pri plote, kde je kopa voľného miesta dohora, ale nie príliš do šírky, a pod prístreškom, kde je síce miesta do šírky dosť, ale sú obmedzené strechou. Ešte sa nerozhodli. Čo však určite vedia, je akú vysokú a širokú búdku potrebuje každý psík, a tiež určite vedia, že šetriť na záhradke miestom je úplne maximálne nutné!

Úloha

Dan a Medvied chcú postaviť panelák pre psíkov. Panelák je zložený z poschodí, ktoré majú tvar obdĺžnika. Každé poschodie má kladnú výšku. Na každom poschodí sa nachádza niekoľko búdiček v rade za sebou, pričom každá búdka má kladnú šírku a výšku. Je zjavné, že poschodie musí byť aspoň tak vysoké ako najvyššia búdka v ňom.

Následne dostanete jeden z dvoch typov otázky. V prvom type máte vypočítať najnižšiu možnú výšku paneláku, pričom šírka paneláku nesmie prekročiť zadanú šírku W . V druhom type máte naopak vypočítať najnižšiu možnú šírku paneláku, pričom výška paneláku nesmie prekročiť zadanú výšku H .

Nakoniec je zadaná postupnosť psíkov. Psíkovia sa musia do domčeka nasťahovať podľa istej hierarchie, a tak im Dan popriradzoval čísla od 0 po $n - 1$. Formálne, pre každých dvoch psíkov $i, j : 0 \leq i < j < n$, musí platiť, že ak psíkovia nebývajú na rovnakom poschodí, tak psík i býva vyššie, a ak bývajú na rovnakom poschodí, tak psík i býva viac vľavo. Neformálne, ak by sme búdky prečítali zhora zľava, prečítame ich ako 0, 1, 2, 3...

Vašou úlohou je v závislosti na otázke vypočítať buď najnižší panelák najviac zadanej šírky, alebo najvyšší panelák najviac zadanej výšky.

Formát vstupu

Na prvom riadku vstupu je číslo n ($1 \leq n \leq 10^5$) udávajúce počet psíkov.

Na druhom riadku vstupu je napísané buď *sirka* alebo *vyska* podľa toho, či sú Dan a Medvied obmedzení šírkou, alebo výškou.

Na treťom riadku sa nachádza číslo W alebo číslo H ($1 \leq W, H \leq 10^{18}$), teda najväčšia šírka/výška, ktorú si môžu Dan a Medvied dovoliť.

Nasleduje n riadkov. Na i -tom z nich sa nachádzajú čísla w_i a h_i ($1 \leq w_i, h_i \leq 10^{12}$) - šírka a výška domčeka pre i -tého psíka. Vždy je garantované, že sa panelák dá postaviť.

Formát výstupu

Vypíš jeden riadok a v ňom jedno číslo, podľa typu vstupu najnižšiu možnú šírku, alebo výšku paneláku.

Hodnotenie

Je 8 sád testovacích vstupov. V jednotlivých sádach platia nasledujúce obmedzenia:

Sada	1	2	3	4	5	6	7	8
$1 \leq N \leq$	11	150	10^3	10^3	10^3	10^5	10^5	10^5
$1 \leq W, H \leq$	100	1000	10^{18}	10^9	10^{18}	10^{18}	10^9	10^{18}
$1 \leq w_i, h_i \leq$	100	1000	10^{12}	10^9	10^{12}	10^{12}	10^9	10^{12}

V tretej a šiestej sade sú všetky otázky typu “sirka”.

Pozor! Teoretické riešenia úlohy môžu získať plný počet iba vtedy, ak ich časová zložitosť závisí iba od n .

Príklady

vstup

```
5
vyska
10
3 3
5 2
2 3
4 3
2 5
```

výstup

```
8
```

Optimálne je dať psíkov na dve poschodia. Prvé bude mať výšku 3 a budú na ňom bývať prví dvaja psíkovia, druhé bude mať výšku 5 a budú na ňom bývať zvyšní psíkovia.

vstup

```
5
sirka
10
3 5
5 4
2 4
9 1
1 2
```

výstup

```
7
```