



# Korešpondenčný seminár z programovania

Leták zimnej časti XL. 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 zadáním 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**. 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 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 1. kola zimnej časti

Termín odoslania riešení tohto kola je pondelok 31. októbra 2022. Doprogramovávanie končí v pondelok 14. novembra 2022.

### 1. Dalo by sa?!

12 b za popis, 8 b za program

“Dalo by sa pohnúť?”

“Kto ti dal vodičák?”

“Slečna ja Vás zveziem, moje auto je dosť veľké...”

“Ludiaaaa? Nevideli ste niekto jeden valec z môjho auta?”

“Čo to bola za značka? Nebolo to *Daj prednosť v jazde*?”

“Odtiahnete ma niekto? Asi som zase prerazil olejovú vaňu...”

“Joooj, posunúť sa, aby som mohol odísť? Dalo by sa?”

“Dávam Vám ešte 2 minúty, a potom sa proste pohnem tým smerom, ako stojím.”

“Vajcia? Ja som naozaj do auta nebalila vajcia! Tie musí mať niekto iný, ja ich určite nemám”

Osadenstvo T2<sup>1</sup> tento hluk spočiatku ignorovalo, ale ako neutíchal, postupne sa preberali z letargie a knedliačenia na gaučoch. Hneď ako vstali a pozreli sa z okna, pochopili čo sa stalo. FKS odchádza na sústredenie. A všetci vedúci išli vlastným autom. Pred sústredkom sa všetci stretli na matfyzе a zastali autom pred okno T2, na jednu dlhú a úzku cestu. Po čase do áut pobalili veci, a ako už chceli všetci odísť, tak zistili, že nikto sa nie je ochotný autom otočiť, každý chce pokračovať tým smerom, ako teraz stojí.

Aby mali na túto situáciu pamiatku, Adam zobral foťák a odfoťil všetky autá, ktoré boli na ceste. Hneď potom sa ale stali naraz dve veci: zapíjala mikrovlnka (a všetci sa k nej otočili) a autá sa pohli (a teda do seba ponarážali). Keďže nikto z osadenstva T2 sa na autá nepozeral, tak nikto nevidel priebeh zrážky, ktorá nastala.

Po čase smútenia (nad tým, že nevideli zrážku, nie nad autami) si ale uvedomili, že možno by sa dalo aspoň z fotky povedať, koľko zrážok sa pred oknom udialo. Vedeli by ste im to zistiť?

### Úloha

Vašou úlohou bude určiť počet zrážok, ktoré nastali pred oknom T2. Pred T2 oknom sa nachádzajú autá, ktoré buď smerujú doľava, stoja, alebo smerujú doprava. Všetky autá, ktoré smerujú doľava alebo doprava sa v rovnakom momente pohli rovnakou rýchlosťou. Vašou úlohou je zistiť počet zrážok, ktoré nastanú. Autá po zrážke zastanú na mieste a ostanú stáť.

Zrážka nastane vtedy, keď hýbajúce sa auto do niečoho narazí (teda ak narazí auto do stojaceho auta, je to jedna zrážka, ak sa čelne zrazia dve autá, čo doteraz išli opačne, sú to 2 zrážky).

### Formát vstupu

Na jedinom riadku vstupu dostanete  $n$  znakov dlhý reťazec. Tento reťazec sa skladá zo znakov > (doprava idúce auto), < (doľava idúce auto), = (stojace auto).

V jednotlivých sadách platia nasledujúce obmedzenia:

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

V prvej a druhej sade sa nevyskytujú stojace autá.

V prvej sade sa všetky autá okrem jedného hýbu rovnakým smerom (napríklad <<<<<, alebo >>>>>).

### Formát výstupu

Vypíšte jediný riadok, počet zrážok, ktoré pred oknami T2 nastanú. Nezabudnite za číslom vypísať znak nového riadku.

<sup>1</sup>KSP miestnosť na matfyzе

## Príklady

|  |        |
|--|--------|
| vstup  | výstup |
| ><><   | 4      |
| vstup  | výstup |
| <=>  | 0      |
| <i>Krajné autá odídu a stredné sa nikam nepohne, takže nenastane žiadna zrážka</i> |        |
| vstup  | výstup |
| >>=<><   | 5      |
| vstup  | výstup |
| >>><><><>>==<>==<><>>><>=><=   | 26     |

## 2. Aerolinkové ceny

12 b za popis, 8 b za program

Nemenovanej spoločnosti sa podarilo navrhnuť dokonalý systém, ako čo najefektívnejšie navrhnuť lety medzi ľubovoľnou skupinou miest tak, aby nebolo treba príliš veľa prestupovať, nemuseli sme chodiť z Bratislavy do Trnavy cez Košice... No, určite si viete aj sami predstaviť mnohé nepríjemnosti, ktoré by mohli nastať, keby lety neboli naplánované rozumne. Čo ale ešte nevymysleli je, že koľko budú stáť jednotlivé lety. Bez veľkého rozmýšľania sa rozhodli priradiť mestám unikátne hodnoty, a cenu letu medzi dvomi mestami určovať ako súčet ich hodnôt. Už teda stačí iba mestám priradiť hodnoty. Lenže ako na to? Hlavným cieľom spoločnosti je samozrejme zarobiť čo najviac. Pomôžte spoločnosti vymyslieť systém ohodnocovania miest tak, aby súčet cien všetkých letov bol čo najväčší.

### Úloha

Vašou úlohou bude priradiť  $n$  mestám rôzne hodnoty od 1 po  $n$  (vrátane), a to tak, aby súčet cien všetkých letov medzi nimi bol čo najväčší. Cenu letu medzi dvomi mestami vypočítame ako súčet hodnôt daných dvoch miest.

### Formát vstupu

Na prvom riadku vstupu sú 2 medzerou oddelené čísla  $n$  a  $m$ , kde  $n$  je počet miest a  $m$  je počet letov medzi nimi. Mestá sú očíslované od 0 po  $n - 1$ . Nasleduje  $m$  riadkov, každý z nich obsahuje 2 medzerou oddelené čísla miest, medzi ktorými sa pravidelne uskutočňujú lety. Rátame s tým, že ak existuje let z A do B, tak existuje aj let z B do A za presne rovnakú cenu, preto s nimi budeme rátať ako s jedným letom a zarátame jeho cenu iba raz (na vstupe nikdy nebude dvojica A B aj B A a taktiež nikdy nebudú na vstupe dva úplne rovnaké lety).

V jednotlivých sadách platia nasledujúce obmedzenia:

| Sada            | 1  | 2      | 3      | 4      |
|-----------------|----|--------|--------|--------|
| $1 \leq n \leq$ | 10 | 500    | 50000  | 100000 |
| $0 \leq m \leq$ | 50 | 125000 | 300000 | 500000 |

### Formát výstupu

Vypíšte jediné číslo, a to najväčší možný súčet cien všetkých letov. Toto číslo môže byť veľké, a teda odporúčame použiť 64 bitovú premennú (`long long` v C++). Nezabudnite za číslom vypísať znak nového riadku.

## Príklady

|                          |        |
|--------------------------|--------|
| vstup                    | výstup |
| 3 3<br>0 1<br>0 2<br>1 2 | 12     |

Keď priradíme mestám hodnoty (`mesto:hodnota`): 0:1, 1:2, 2:3, tak potom súčet cien všetkých letov by bol:  $(1 + 2) + (1 + 3) + (2 + 3) = 12$ . Môžeme si ale všimnúť, že v tomto prípade by sme pre ľubovoľné priradenie dostali rovnaký súčet.

vstup

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

výstup

```
13
```

Pre priradenie: 0:3, 1:4, 2:2, 3:1 získame takýto súčet:  $(3 + 4) + (4 + 2) = 13$ . Ak by sme ale zvolili napríklad priradenie: 0:1, 1:2, 2:3, 3:4, tak by bol súčet cien všetkých letov menší, a to:  $(1 + 2) + (2 + 3) = 8$ .

vstup

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

výstup

```
37
```

### 3. Lietajúce Taniere

12 b za popis, 8 b za program

Pozemšťanské Národné múzeum v Sekulách vystavuje skutočný skvost: Najvzácnejšiu hlásku galaxie<sup>2</sup>. Táto hláška je lokálny endemit – nevyskytuje sa vo fonológii jazyka žiadnej inej planéty, a preto má obrovskú hodnotu. A práve preto bola v piatok cez obedovú prestávku ukradnutá rafinovaným páchatelom. Nie je ním nik iný, ako interstelárny kriminálnik Okáň Hruškový, prezývaný “Bzučiak”. Samotná krádež bola pre neho hračkou, skutočná skúška jeho schopností nastáva až teraz: s ukradnutou hláskou sa musí dostať na najbližšiu matriku, kde si zmení meno na Okáň, čím zmätie agentov zákona, zahladí všetky stopy po svojej kriminálnej histórii a šťastný odíde do dôchodku. Samozrejme, poletí lietajúcim tanierom<sup>3</sup>.

Má to však háčik: Medzi Pozemšťanským Národným múzeom v Sekulách a Sekulským Matričným úradom nejde žiaden priamy let, preto musí Bzučiak prestúpiť na Hlavnej Tanierovej Stanici Sagittarius A.

Okrem toho je problémom, že galaktická polícia má agentov zákona všade. Normálne síce pre Bzučiaka žiaden problém nepredstavujú<sup>4</sup>, pokiaľ ale poletí rovnakým letom ako jeden z nich, môže sa s novým menom rozlúčiť.

Ako nový zamestnanec IT oddelenia galaktickej polície ste dostali časy odletov všetkých liniek, spolu s prídelenými financiami, za ktoré môžete nakúpiť niekoľkým policajtom letenky, aby ste Bzučiakovi znemožnili využitie týchto letov. Vašou úlohou je Bzučiakovi cestu čo najviac zneprijemniť, pokiaľ možno úplne znemožniť.

#### Úloha

Máte dané časy odchodov všetkých  $n$  letov z Národného múzea do Tanierovej Stanice, ako aj ich dĺžku  $d_a$ , ktorá je, samozrejme, pre všetky rovnaká. Rovnako máte dané aj odchody všetkých  $n$  letov z Tanierovej Stanice na Matričný úrad, a ich dĺžku  $d_b$ . Máte peniaze na zablokovanie  $k$  letov – určite, v akom najneskoršom čase môžeme Bzučiaka prinútiť dostať sa na Matričný úrad, respektíve či vieme Bzučiakovi v jeho dosiahnutí úplne zabrániť.

Bzučiak môže prestúpiť medzi dvoma letmi, pokiaľ čas príchodu prvého nie je vyšší ako čas odchodu druhého, pričom dĺžka letu je definovaná ako rozdiel v časoch jeho príchodu a odchodu.

#### Formát vstupu

Na prvom riadku vstupu sa nachádzajú 4 celé čísla  $n, d_a, d_b, k$  - počet letov na každej linke, dĺžky letov prvej a druhej linky, a počet lístkov, ktorý si môžeme dovoliť kúpiť, pričom vždy platí  $1 \leq d_a, d_b$  a  $1 \leq k \leq 2n$ .

Nasledujú dva riadky. Na prvom z nich je  $n$  rôznych celých čísel  $a_0$  až  $a_{n-1}$ , ktoré označujú časy odchodov letov prvej linky. Na druhom z nich je  $n$  rôznych celých čísel  $b_0$  až  $b_{n-1}$ , ktoré označujú časy odchodov letov druhej linky. Platí, že oba riadky sú zoradené vo vzostupnom poradí, a okrem toho všetky časy odletov  $x$  spadajú do nasledovného rozmedzia:  $1 \leq x \leq 10^9$ .

#### Formát výstupu

Na jediný riadok výstupu vypíšete jediné číslo – najneskorší čas, v akom môžeme Bzučiaka donútiť doraziť do cieľovej zastávky. Pokiaľ je možné Bzučiakovi zabrániť v dosiahnutí cieľa úplne, vypíšete číslo  $-1$ . Nezabudnite na konci vypísať symbol konca riadku.

<sup>2</sup>Lokálne sa zaznačuje písenkom ä.

<sup>3</sup>Nemýliť si s morským tanierom.

<sup>4</sup>Dôvod, prečo Bzučiaka polícia nemôže chytiť mimo lietajúceho taniera, je úplne očividný a netreba ho vysvetľovať.

## Hodnotenie

Sú 4 sady vstupov:

- V prvej sade platí, že  $1 \leq n \leq 100$ ,  $d_a = d_b = 1$  a obe linky majú rovnaké časy odletov, teda  $a_i = b_i$  pre všetky  $0 \leq i \leq n - 1$ .
- V druhej sade platí, že  $1 \leq n \leq 10\,000$ .
- V tretej sade platí, že  $1 \leq n \leq 100\,000$ .
- V štvrtej sade platí, že  $1 \leq n \leq 1\,000\,000$ .

## Príklad

| vstup                              | výstup        |
|------------------------------------|---------------|
| <pre>4 3 5 3 1 2 3 6 6 7 8 9</pre> | <pre>14</pre> |

Zablokujeme napríklad prvé tri lety z Národného múzea. Bzučiak nebude mať inú možnosť, ako využiť ten posledný, s odchodom v čase 6. V Tanierovej Stanici bude v čase 9, stihne rýchlo nastúpiť na posledný let na Matričný Úrad, ktorý k nemu dorazí v čase 14. Alternatívne sme mohli zablokovať prvé tri lety druhej linky a dosiahnuť rovnaký výsledok.

| vstup   | výstup        |
|---|---------------|
| <pre>6 2 1 4 1 3 5 7 9 11 3 4 5 6 10 13</pre> | <pre>-1</pre> |

V tomto prípade vieme Bzučiaka od Matričného Úradu úplne odrezat, napríklad zablokovaním prvých dvoch letov z Národného múzea a posledných dvoch letov z Tanierovej Stanice. Bzučiak odletí najskorším letom s odchodom v čase 5, v Tanierovej Stanici bude v čase 7, a nestíha už žiaden nezablokovaný let ďalej.

## 4. Opačné čítanie

12 b za popis, 8 b za program

S jeseňou okrem chladného počasia a padania listov prichádza aj nový školský rok a Adam musel opäť nastúpiť do školy. Keďže sa do školy vrátiť nechcel, vymýšľal všelijaké výhovorky, aby mohol zostať doma (napr. strčil teplomer do teplého čaju; presviedčal rodičov, že je stále august; schoval sa pod posteľ; ...). Tie sa mu však neprepiekli a do školy išiel aj tak.

Ako tam sedel, už na druhej vyučovacej hodine, uvedomil si, že to napokon nie je až také zlé. Čo ale čert nechcel, cez veľkú prestávku si spolužiaci začali robiť srandu z jeho štíhlejšej stavby tela. Na lavici mu pristál papierik so slovami **Madam Adam**. Adama to trošku rozhodilo a spýtal sa učiteľky, či by sa dalo ísť domov. Učiteľka myslíac, že sa pýtal na záchod, ho s úsmevom na perách pustila.

Po príchode domov si ešte raz otvoril papierik s odkazom. Ako sa naň pozeral, uvedomil si zaujímavú vec – **madamadam** sa číta rovnako spredu aj zo zadu. Takáto vlastnosť sa Adamovi veľmi zapáčila a vymyslel si takých slov viac. Ba dokonca, vymyslel ku takýmto slovám zaujímavú úlohu.

### Úloha

Adam vymyslel  $n$  stringov s  $k$  znakmi, ktoré sa čítajú rovnako spredu aj zo zadu. String obsahuje znaky  $a-z$ ,  $A-Z$  a  $0-9$ . Okrem toho, v každom stringu je každý unikátny znak **maximálne dvakrát**. Vašou úlohou je zistiť, koľkými rôznymi spôsobmi sa dajú vymazať znaky zo stringu, aby sa nečítal rovnako spredu aj zo zadu.

### Formát vstupu

Na prvom riadku vstupu je číslo  $n$ , ktoré reprezentuje počet otázok. Potom nasleduje  $n$  riadkov a na každom z nich je string dĺžky  $k$ .

### Formát výstupu

Na každý riadok vypíšte číslo, koľkými rôznymi spôsobmi vieme vymazať znaky v stringu, aby sa nedal prečítať rovnako spredu aj zo zadu. Keďže odpoveď môže byť veľmi veľká, vypisujte ju modulo  $10^9 + 7$ .

### Obmedzenia

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

| Sada            | 1  | 2   | 3     | 4      |
|-----------------|----|-----|-------|--------|
| $1 \leq n \leq$ | 10 | 100 | 1 000 | 10 000 |
| $1 \leq k \leq$ | 10 | 25  | 75    | 125    |

### Príklad

vstup

1  
AbbA

výstup

6

Po vymazaní znakov zo stringu *AbbA* je týchto 6 možných stringov, ktoré sa nečítajú rovnako spredu aj zo zadu: *Abb*, *Ab*, *Ab*, *bbA*, *bA*, *bA*. Na prvý pohľad sa zdá, že stringy *Ab* a *Ab* sú rovnaké, lenže jeden string obsahuje znak *b* na prvom indexe a druhý obsahuje znak *b* na druhom indexe, čiže sú rôzne. Tak isto aj *bA* a *bA*.

vstup

2  
DaL00LaD  
bysAsyb

výstup

210  
98

Zo stringu *DaL00LaD* sa dajú vymazať znaky 210 spôsobmi a zo stringu *bysAsyb* sa dajú vymazať znaky 98 spôsobmi, aby sa nečítali rovnako spredu aj zo zadu.

## 5. Bezkonkurenčná manufaktúra

12 b za popis, 8 b za program

V dnešnej zúboženej ekonomike sa možno iba ťažko na niečo spoľahnúť. Jednou výnimkou tohoto pravidla je však nemenný dopyt po sršňom mede. Ten ako jediný zostal neovplyvnený, po tom, čo nedávne udalosti postihli už aj ziskovosť produkcie pavúčieho medu.

Vašej sršnej farme sa už dlhšiu dobu darí náramne dobre. Morálka je vysoká, produkcia je bezproblémová a výrobok je kvalitný. Sršne sú vďaka desiatkam hodín nadštandardného tréningu schopné a vysoko zorganizované. Na rozdiel od včiel majú sršne vďaka svojej vyššej inteligencii rádovo väčší potenciál, ktorý má v kombinácii s poctivým prvotriednym tréningom za následok pedantne efektívnu Lean-Agile manufaktúru s rozsiahlou hierarchiou vedenia. Primitívny pôvod včelieho medu je v porovnaní na smiech.

### Úloha

Výroba špičkového sršnieho medu je vysoko technická záležitosť založená na častej komunikácii. Tá prebieha *iba* v rámci hierarchie. Teda každý sršeň vie komunikovať iba so svojim priamym nadriadeným alebo podriadenými, pričom na umožnenie komunikácie ľubovoľnej dvojice sršňov bol zavedený systém preposielania správ. Sršnia hierarchia samozrejme umožňuje komunikáciu ľubovoľným dvom sršňom a to dokonca unikátnym najkratším spôsobom. V hierarchii sa teda nevyskytujú cykly, keďže v internom testovaní sa ukázalo, že ich prítomnosť síce občas skrátila čas dodania správy, avšak pridaná komplexita narúšala sršňom pracovný flow.

Tento systém doteraz fungoval bezchybne. Máte však obavy o jeho dlhodobom vplyve na niektorých jedincov. V hierarchii sa vyskytuje niekoľko sršňov, ktoré slúžia ako nevyhnutný sprostredkovateľ komunikácie veľkého množstva dvojíc. Dokonca ste zistili, že v nej existuje jeden sršeň, ktorý je najvyťaženejší zo všetkých a o ktorého máte preto najväčšie obavy. Ste teda ochotní umožniť komunikovať jednej ľubovoľnej dodatočnej dvojici sršňov, aby ste ho aspoň trochu odbremenili. Zaujímalo by vás, koľko dvojíc využíva tohoto sršňa ako sprostredkovateľa a koľko najmenej dvojíc ho bude stále nutne využívať po tom, čo umožníte komunikáciu jednej dodatočnej dvojici (výber dvojice nechávame na vaše uváženie). Vašou úlohou bude túto analýzu efektívne vykonať na všetkých manufaktúrach vašej farmy.

### Formát vstupu

Na vstupe bude na prvom riadku číslo  $2 \leq N \leq 100001$  označujúce počet sršňov v práve analyzovanej manufaktúre. Nasleduje  $N - 1$  riadkov popisujúcich hierarchiu sršňov - na každom z nich dve čísla  $0 \leq A, B < N$  symbolizujúce susednosť sršňov  $A$  a  $B$  v rámci hierarchie. Hierarchia na vstupe je súvislá a bez cyklov. Sršeň 0 je vedúcim manufaktúry a existuje práve jeden najvyťaženejší sršeň.

### Formát výstupu

Na výstup vypíšete na práve jeden riadok práve dve celé čísla v desiatkovej sústave oddelené práve jednou medzerou. Prvé číslo je počet dvojíc sršňov, ktoré nutne komunikujú cez najvyťaženejšieho sršňa a druhé číslo je to isté, avšak v prípade umožnenia komunikácie jednej dodatočnej dvojici.

## Príklady

vstup

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

výstup

```
11 5
```

*Sršeň 2 je navyžťaženejší. Po tom, čo umožníme komunikovať napríklad sršňom 0 a 6, už bude sršeň 2 nutne sprostredkovávať iba komunikáciu sršňa 3 so všetkými ostatnými (avšak pri komunikácii sršňov 2 a 3 neslúži sršeň 2 ako sprostredkovateľ).*

## 6. Yha to je šuter

12 b za popis, 8 b za program

Bubu sa jedného dňa prechádzal po lese, keď mu do oka padol šuter. Chvalabohu, iba metaforicky, nie doslova. Šutrák to bol riadny, čo Bubu potešilo, až radostne zýýkol (Yyyyyy!). No keď si Bubu uvedomil, že má tento balvan v ceste, radosť ho prešla. Pobral sa ho obísť, no hneď za šutrom, na čistine, mu jeden strom, starý hodne, zahatil, aby nezaspal, preventívne cestu tým, že sa vyvalil. Bubu sa nenechal odradiť a aj túto prekážku obišiel. Potom, čo obišiel tento strom, sa mu kdesi v podlesí zjavili v ceste dva vresy (a to riadne prerastené). Za mohutným sudom od kapusty (ktorý musel tiež chudák obchádzať) ležali vyvalené 4 bresty. A tak ďalej, (no viete si to domyslieť), no a keď bol koniec prechádzky, tak mu do cesty vošli sanitky.

Tak s potom až na brade, vo vysilenej nálade, Bubu dokončil svoje trápenie počnuté kusom žuly. Teraz hľadá optimálne riešenie ako obísť skaly.

### Úloha

Vašou úlohou je nájsť najkratšiu trasu z pozície udanej začiatočnými koordinátami na pozíciu udanú konečnými koordinátami, ktorá neprechádza žiadnou z popísaných prekážok. Prekážky sú buď úsečkami, alebo konvexnými mnohoúhľovníkmi.

Vo všeobecnosti platí, že sa tieto prekážky nedotýkajú. Inak povedané, začiatok a koniec žiadnej úsečky, vrátane vrcholov mnohoúhľovníkov, nikdy neležia uprostred inej úsečky. (V niektorých sadách sa ale úsečky môžu pretínať a zdieľať koniec/začiatok)

Pozn.: Cesta môže prechádzať pozdĺž prekážky, aj cez jej začiatok a koniec, ale nie cez samotnú prekážku, čo v prípade mnohoúhľovníkov znamená, že môže ísť po obvode ale nie cez ich vnútro.

### Formát vstupu

Na prvom riadku vstupu sú štyri čísla  $x_s, y_s, x_e, y_e$  popisujúce koordináty začiatku  $(x_s, y_s)$  a konca  $(x_e, y_e)$  trasy, ktorú chce Bubu prejsť.

Druhý riadok obsahuje jedno číslo  $n$ , počet prekážok.

Nasleduje  $2n$  riadkov, ktoré po pároch popisujú prekážky na trase. Riadok číslo  $2k$  obsahuje jedno číslo  $m_k \geq 2$ , počet bodov popisujúcich  $k$ -tú prekážku. V prípade, že  $m_k = 2$ , je prekážka úsečkou, v opačnom prípade je mnohoúhľovníkom.

Riadok číslo  $2k + 1$  obsahuje  $2m_k$  čísel vo formáte  $x_0, y_0, \dots, x_{m_k-1}, y_{m_k-1}$ , ktoré popisujú koordináty prekážky, v poradí.

V jednotlivých sadách platia nasledujúce obmedzenia:

| Sada                   | 1, 2 | 3, 4 | 5, 6 | 7, 8 |
|------------------------|------|------|------|------|
| $1 \leq \sum m_i \leq$ | 20   | 60   | 150  | 200  |

V sadách 1, 2, 3, 4 môžete predpokladať, že sa žiadne dve úsečky nepretínajú. V sadách 1, 3, 5, 7 navyše platí  $m_i = 2$ , vo všeobecnosti platí  $m_i \leq 10$ .

### Formát výstupu

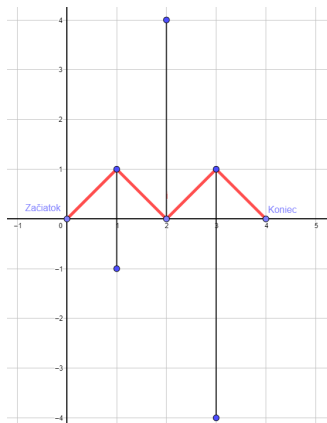
Vypíšte dĺžku najkratšej novej trasy zo začiatku na koniec. Váš výsledok by mal byť najviac  $10^{-6}$  rozdielny od očakávaného výsledku.



## Príklady

vstup

```
0.0 0.0 4.0 0.0
3
2
1.0 -1.0 1.0 1.0
2
2.0 4.0 2.0 0.0
2
3.0 1.0 3.0 -4.0
```



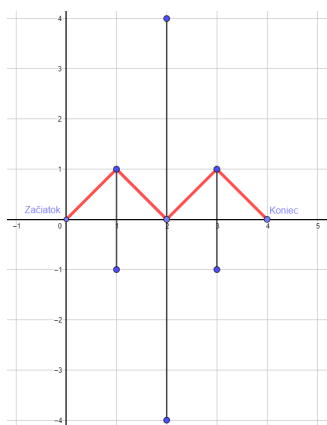
Prekážky sú čierne úsečky s vyznačeným koncom a začiatkom, optimálna trasa je červenou.

výstup

```
5.656854249492381
```

vstup

```
0.0 0.0 4.0 0.0
4
2
1.0 -1.0 1.0 1.0
2
2.0 4.0 2.0 0.0
2
2.0 0.0 2.0 -4.0
2
3.0 1.0 3.0 -1.0
```



V prípade, že dve úsečky zdieľajú jeden koniec, môže tade prechádzať trasa.

výstup

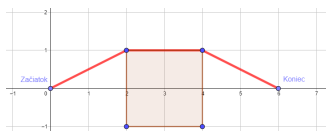
```
5.656854249492381
```

vstup

```
0.0 0.0 6.0 0.0
1
4
2.0 2.0 2.0 -2.0 4.0 -2.0 4.0 2.0
```

výstup

```
7.656854249492381
```



Podobne ako vyššie, štvorcová prekážka je hnedou.

## 7. Srny, Viki, Viki a iné divé veci

12 b za popis, 8 b za program

Tento príbeh je čisto fiktívny, a akákoľvek podobnosť s udalosťami Suši chaty je čisto náhodná<sup>5</sup>.

Miško je už dve hodiny bez jedla, a teda mu ostáva už len zopár minút života, pokiaľ nebude nakŕmený. Našťastie majú Viki a Viki dobré srdcia a rozhodli sa Miškovi život zachrániť. Vzali vysoký kaleráb, a položili ho na krájaciu dosku naležato. Vzniknutý široký kaleráb rozdelili na dva široké kusy kalerábu. Každý z nich si môžeme predstaviť ako rad čísel – každé číslo je kusom šupy širokého kusu kalerábu, a označuje počet listov na danom kuse šupy (to bude dôležité neskôr). Viki a Viki šúpu kaleráb tak, že každý má vlastný široký kus kalerábu. Následne opakujú jednoduchý proces: Viki aj Viki zo svojho kusu ošúpe z konca nejaký široký kus šupy a následne oba ošúpané kusy šupy zahodia do koša. Tento proces opakujú, kým obe široké kusy kalerábu neošúpu celé.

Ako to už v takýchto situáciách býva, plány vedúcich boli zrušené matkou prírodou. V okolí Suši chaty totiž žijú divé srny, ktoré nedokážu odolať chuti listov širokého kalerábu. Vždy, keď Viki a Viki zahodia dvojicu kusov šupy do koša, v koši sa objaví niekoľko nových sŕn, podľa jednoduchého vzorca:

Nárast populácie divých sŕn v koši sa dá vypočítať ako súčin rozdielu počtu listov na prvom kuse šupy a šírky prvého kusu šupy a rozdielu počtu listov na druhom kuse šupy a šírky druhého kusu šupy.

Miško je príliš hladný, Viki príliš lenivý a Viki nie je vedúca KSP. Je teda len na vás, aby ste zistili, ako treba obe časti širokého kalerábu šúpať tak, aby po ich ošúpaní bolo v koši čo najmenej sŕn.

### Úloha

Na vstupe dostanete dve sekvencie kladných celých čísel – počty listov na šupách oboch kusov kalerábu. V jednom kroku Viki a Viki zvolia dve kladné čísla  $x, y$ . Z prvej sekvencie zmažú z konca  $x$  čísel, z druhej  $y$ , a následne sa v koši objaví nasledovný počet sŕn:  $(S - x) * (Z - y)$  Pričom  $S$  je súčet  $x$  zmazaných čísel z prvej sekvencie a  $Z$  je súčet  $y$  zmazaných čísel z druhej sekvencie. Vašou úlohou je určiť, koľko najmenej sŕn sa môže v koši objaviť v procese šúpania kalerábu, kým kaleráb ošúpeme celý, ak šúpeme optimálne.

### Formát vstupu

Na prvom riadku vstupu sú 2 čísla  $n, m$  – šírky kusov kalerábu. V ďalšom riadku nasleduje  $n$  čísel – počty listov na Vikiho kuse kalerábu. V ďalšom riadku je  $m$  čísel – počty listov na Vikiho kuse kalerábu.

### Formát výstupu

Na jediný riadok výstupu vypíšte jedno číslo – počet sŕn v koši, ak Viki a Viki šúpali kaleráb optimálne. Nezabudnite na koniec riadku.

### Hodnotenie

Sú 4 sady vstupov, za každú sú 2 body. Vo všetkých vstupoch platí, že  $n, m \geq 1$  a zároveň na každej šupe je aspoň 1 list. Na žiadnej šupe nie je viac ako 1000 listov. Ďalšie obmedzenia si môžete pozrieť v tabuľke.

| sada | $n, m \leq$ |
|------|-------------|
| 1.   | 6           |
| 2.   | 100         |
| 3.   | 300         |
| 4.   | 1000        |

<sup>5</sup>To je čosi ako čarodejnicky sabat, ale namiesto čarodejnic sú tam vedúci Súfaže v Šifrovaní (malý rozdiel).

## Príklady

vstup

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

výstup

```
2
```

Najprv Viki aj Viki ošúpu 1 šupu zo svojho kusu kalerábu. Pritom sa v koši objavia  $(3 - 1) \times (2 - 1) = 2$  divé srny. Potom Viki ošúpe 2 šupy zo svojho kusu kalerábu, a Viki jednu šupu zo svojho kalerábu. Pritom sa v koši objaví  $(3 - 2) \times (1 - 1) = 0$  divých srn. Za celý čas sa teda v koši objavia 2 divé srny.

## 8. Abnormálne Veľký Hotel

12 b za popis, 8 b za program

### Úloha

Pridelovač<sup>6</sup> pracuje v hoteli s  $n$  izbami číslovanými od 0 po  $n - 1$  vrátane.

Pridelovač dostáva dva druhy požiadaviek:

- check-in požiadavky, ktoré obsahujú číslo  $x$  – počet miestností potrebný pre skupinu hostí. Pridelovač týchto hostí ubytuje do  $x$  po sebe idúcich voľných izieb. Pridelovač vždy vyberie taký usek izieb, ktorý minimalizuje minimálne číslo izby zo všetkých čísel izieb v úseku.
- check-out požiadavky, ktoré obsahujú číslo  $i$  – index skoršej check-in požiadavky. Check-in požiadavky indexujeme od 0 a do indexovania nerátame check-out požiadavky. Skupina hostí ubytovaná v  $i$ -tej check-in požiadavke uvoľní svoje miestnosti a opustí hotel.

Je zaručené, že Pridelovač bude schopný naplniť všetky požiadavky.

Vašou úlohou je vypísať pre každú check-in požiadavku číslo najmensej izby z úseku miestností pridelených skupine hostí.

### Formát vstupu

Na prvom riadku vstupu sú dve medzerou oddelené prirodzené čísla  $n$ ,  $q$  – počet izieb v hoteli a počet požiadaviek.

Nasleduje  $q$  riadkov. Každý riadok obsahuje jednu požiadavku. Check-in požiadavky majú tvar **I**  $x$ ,  $x$  – počet miestností potrebný pre skupinu hostí. Check-out požiadavky majú tvar **O**  $i$ ,  $i$  – index skoršej check-in požiadavky.

### Formát výstupu

Pre každú check-in požiadavku vypíšte na jeden riadok jediné číslo – číslo najmensej izby z úseku miestností pridelených skupine hostí.

### Hodnotenie

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

| Sada            | 1      | 2      | 3       | 4         |
|-----------------|--------|--------|---------|-----------|
| $1 \leq n \leq$ | $10^3$ | $10^9$ | $10^9$  | $10^{18}$ |
| $1 \leq q \leq$ | 10     | 5 000  | 300 000 | 600 000   |

### Príklad

vstup

```
9 7
I 3
I 3
O 0
I 2
I 2
I 1
I 1
```

výstup

```
0
3
0
6
2
8
```

<sup>6</sup>Ujo, ktorého úlohou je ubytovávať a odubytovávať hostí

Reprezentujme hotel stringov znakov, kde  $i$ -ty znak reprezentuje stav  $i$ -tej izby. Spočiatku je hotel prázdny ----- . Príde skupina troch hostí 000----- . Príde ďalšia skupina troch hostí 000111--- . Odubytuje sa skupina hostí 0 ---111--- . Príde skupina 2 hostí 22-111--- a na to ďalšia 22-11133- . Nakoniec príde jeden človek 22411133- a za ním ďalší 224111335 .