



# Korešpondenčný seminár z programovania

Leták zimnej časti XXXVII. 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 neprezdádzaj 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 neapísali 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).

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 21.10.2019. Doprogramovanie končí v podnelok 4.11.2019.

### 1. Dvojičky

12 b za popis, 8 b za program

Ísť na rodinnú oslavu asi nebol až taký dobrý nápad ako by sa mohlo zdať. Nie že by koláče neboli fajn, alebo by si nemal rád svojich čudných príbuzných, ale tá tvoja rozvetvená rodina... Teda, aj tá by bola v pohode, ak by ste nemali v rodine toľko dvojičiek.

Našťastie, nikdy si nemusel vedieť mená tvojich príbuzných, ich vek, a ani to, či sú dvojičky... Až doteraz... Prikývol si sestre tety vnučky krstnej tvojej starej mamy z druhého kolena, že jej pomôžeš s jej nečakaným nápadom, odovzdávaním ceny pre najstaršie dvojičky z minulej oslavy.

Sestra tety vnučky krstnej tvojej starej mamy z druhého kolena cenu odovzdala, nechala ti fotku z predošlej oslavy a išla sa najesť koláčov a kapustnice. Aj ty by si sa rád najedol, no po chvíli ťa, ako jej asistenta, začali prenasledovať nespokojné dvojičky, ktoré cenu nedostali s tým, aby si im dokázal, že cena bola odovzdaná správnemu páru dvojičiek.

Chcú teda, aby si im povedal, koľko párov dvojičiek bolo na oslave, a aké staré boli najstaršie dvojičky... Na tvojom mieste by som im rýchlo, ale najmä správne, odpovedal. Chceš od nich predsa dostať na Vianoce nejaké darčeky...

#### Úloha

Dostanete veku ľudí, ktorí sú zachytení na fotke. Týchto ľudí je  $n$ . Na oslave sa nachádzalo  $n+1$  ľudí. Človek, ktorý fotku fotil, nebol na fotke. Ľubovoľní dvaja ľudia, okrem dvojice tvoriacej dvojičku, sú rozdielne starí.

Vypíšte, koľko najviac párov dvojičiek mohlo byť na oslave, a vek najstaršieho možného páru dvojičiek, ktorý bol na oslave, pričom môžete predpokladať, že na oslave sa nachádzal aspoň jeden pár dvojičiek.

#### Formát vstupu

Vstup tvorí dvojica riadkov, ktorá popisuje fotku z oslavy.

V prvom riadku sa nachádza číslo  $1 \leq n \leq 100\,000$ , počet ľudí na fotke.

Na druhom riadku sa nachádza  $n$  čísel  $a_i$  oddelených medzerami,  $1 \leq a_i \leq 200\,000^1$ , ktoré označujú veku ľudí na fotke.

#### Formát výstupu

Vypíšte dve, medzerou oddelené čísla. Koľko najviac párov dvojičiek mohlo byť na oslave, a najväčší možný vek páru dvojičiek, ktorý bol na oslave.

#### Príklad

vstup

```
8
12 12 16 102 47 16 102 47
```

výstup

```
4 102
```

### 2. Všetkých nás nezastavia!

12 b za popis, 8 b za program

Všetci vieme, že mimozemšťania existujú. Dokonca už aj navštívili Zem. Presnejšie povedané, havarovali do nej. Toto miesto je bežne známe ako [Area 51](https://en.wikipedia.org/wiki/Area_51)<sup>2</sup>. Žiaľ, mimozemšťanov sa zmocnilo americké letectvo. Oni si ale zaslúžia byť voľní! Poďme ich zachrániť!

Dvojičky Matty a Robrets pomocou [eventu na Facebooku](#)<sup>3</sup> zohnali milióny ochotných ľudí. Ich tímu sa už podarilo preniknúť do tajnej americkej základne, stačí im už len odtiaľ vyniesť mimozemšťanov. Každý človek

<sup>1</sup>Okrem dvojičiek nie je vo vašej rodine výnimočná ani dĺžka.

<sup>2</sup>[https://en.wikipedia.org/wiki/Area\\_51](https://en.wikipedia.org/wiki/Area_51)

<sup>3</sup>[https://en.wikipedia.org/wiki/Storm\\_Area\\_51,\\_They\\_Can't\\_Stop\\_All\\_of\\_Us](https://en.wikipedia.org/wiki/Storm_Area_51,_They_Can't_Stop_All_of_Us)

unesie maximálne dvoch mimozemšťanov – jedného v každej ruke. Ale s mimozemšťanom pod pazuchou sa ťažšie beží. Mimozemšťanov preto potrebujú rozdeliť tak, aby ten, čo bude niesť najväčšiu záťaž, toho niesol čo najmenej. Preniknutie do základne bolo veľmi náročné, a počítať sa im už nechce. Preto im musíte pomôcť vy!

## Úloha

V Area 51 je  $m$  mimozemšťanov. Na to, aby sme ich zachránili, máme  $n$  ľudí. Každý človek unesie maximálne dvoch mimozemšťanov. Každý mimozemšťan má nejakú konkrétnu hmotnosť  $H_i$  udanú v miligramoch.

Mimozemšťanov musíme rozdeliť medzi jednotlivých ľudí tak, aby maximálna hmotnosť ktorú má niektorý človek niesť, bola najmenšia možná.

### Formát vstupu

V prvom riadku vstupu sú čísla  $n$  a  $m$  ( $1 \leq m \leq 2n \leq 200\,000$ ) udávajúce počet ľudí a mimozemšťanov.

V druhom riadku je  $m$  čísiel  $H_i$  ( $1 \leq H_i \leq 1\,000\,000\,000$ ) udávajúcich hmotnosti mimozemšťanov v miligramoch.

### Formát výstupu

Pre každého z  $n$  ľudí vypíšte jeden riadok obsahujúci dve čísla – hmotnosti mimozemšťanov v jednotlivých rukách. Ak má človek niektorú ruku prázdnu, vypíšte hmotnosť 0.

*Môžete vypísať ľubovoľné správne riešenie.*

### Príklady

vstup

```
3 4
5 1 6 7
```

výstup

```
7 0
0 6
1 5
```

*Najťažší náklad nesie prvý človek, a to 7mg*

vstup

```
3 6
1 2 2 3 3 4
```

výstup

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

*Všetci nesú rovnakú hmotnosť, a to 5mg.*

## 3. Obojstranná čokotuba

12 b za popis, 8 b za program

Samo a Jano sú influenceri. Od čokofirmy dostali masívnu reklamnú obojstrannú tubu rôznofarebných čokoládiek pre dvojčičky. Vraj aby sa odfotili, ako im chutí a aké skvelé to je, že sa nemusia hádať, kto si vytiahne čokoládu prvý. Každý vie, že rôzne farby čokoládiek znamenajú rôzne obsiahnuté živiny. Keďže Samo a Jano chcú všetkým ukázať, ako vyvážene sa stravujú, zráтали si, koľko ktorých čokoládiek chcú zjesť. Dve purpurové, šesť egyptských modrých, jedna nachová, tri indigové, sedem sivých. . .

Tuba s čokoládkami je otvorená z dvoch strán. Samo a Jano začali zbesilo vyťahovať čokošky z jednej strany. Keď nejakú vytiahnú, tak ju rovno aj zjedia, aj keď nepatrí do ich požiadaviek na vyváženú stravu. Keby ju nezjedli, určite by sa niekde stratila. To by bola škoda. Rýchlo si však uvedomili, že čím viac nechcených čokoládiek zjedia, tým horšiu mienku o nich budú mať ich sledovatelia.

Nuž, Samo začal hútať, koľko čokoládiek z ktorej strany to majú vlastne vybrať, aby sa najedli čo najmenej, ale aby zároveň uspokojili svoje potreby vyváženej stravy. Ak budú musieť tých zlých čokolád zjesť priveľa, nebudú ani chcieť pristúpiť na takúto ponuku sponzora. A podobných ponúk určite príde ešte veľa. Hodil by sa im preto program, ktorý pre danú tubu čokoládiek a ich potreby vyváženej stravy vypočíta, koľko najmenej čokoládiek budú musieť vytiahnúť, aby sa dostali k tým, čo chcú. Pomôžete im udržať priazeň publika?

## Úloha

Tuba je priehľadná a otvorená z dvoch strán. Vnútri sú čokoládky naskladané v rade vedľa seba. Samo a Jano chcú vybrať z tuby **aspoň**  $a_1$  čokoládiek farby  $f_1$ , **aspoň**  $a_2$  čokoládiek farby  $f_2$ , . . . , **aspoň**  $a_m$  čokoládiek farby  $f_m$ .

Vašou úlohou je zistiť, koľko najmenej čokoládiek musia dokopy z tuby vybrať tak, aby splnili svoje požiadavky. V každom momente môžu vybrať buď čokoládku úplne zľava, alebo čokoládku úplne sprava.

## Formát vstupu

Na prvom riadku dostanete čísla  $n$  - počet všetkých čokoládiek v tube,  $m$  - počet Samových a Janových požiadaviek a  $f$  - počet rôznych farieb čokoládiek, ktoré sa môžu vyskytovať v tube. Platí, že  $1 \leq f \leq n \leq 500\,000$  a  $0 \leq m \leq f$ .

Na druhom riadku je  $n$  čísiel  $c_1$  až  $c_n$  - farby čokoládiek tak, ako sú v tube zľava doprava. Farby majú čísla od 1 po  $f$ . To však neznamená, že sa na vstupe musia nutne vyskytovať všetky farby od 1 po  $f$ .

Nasleduje  $m$  riadkov. V  $i$ -tom z nich sú čísla  $a_i$  a  $f_i$ . Tie znamenajú, že Samo a Jano chcú z tuby vybrať **aspoň**  $a_i$  čokoládiek farby  $f_i$ . Môžete predpokladať, že všetky  $f_i$  sú navzájom rôzne a tiež, že pre zadanú tubu sú všetky požiadavky vždy splniteľné.

## Formát výstupu

Na výstup vypíšete jedno číslo - koľko najmenej čokoládiek musia Samo s Janom z tuby vybrať, aby splnil všetky svoje požiadavky.

## Príklady

| vstup                                | výstup       |
|--------------------------------------|--------------|
| <pre>6 2 3 1 2 2 1 3 2 1 1 3 2</pre> | <pre>4</pre> |

*Naši súrodenci chcú jednu čokoládu farby 1 a tri čokolády farby 2. Môžu teda zobrať napríklad 3 čokolády zľava a jednu sprava.*

| vstup                          | výstup       |
|--------------------------------|--------------|
| <pre>5 1 3 1 2 2 3 1 1 2</pre> | <pre>2</pre> |

*Teraz chcú iba jednu čokoládu farby 2. Tá ale nie je na kraji, tak sa k nej musia dostať cez najľavejšiu čokoládu farby 1.*

## 4. Jeden smer

12 b za popis, 8 b za program

“Ideme dobrým smerom?” opýtal sa Maťko Kubka. Kubko mal kompas a Maťko mal mapu. “Ideme niekde medzi severom a východom.” povedal Kubko. Maťko sa pozrel na mapu a povedal: “Veľmi si mi nepomohol. Ak ideme skôr k severu, tak to minieme o desať kilometrov a ak viac k východu, tak možno aj o päť.” Kubko sa pozrel na kompas a zamyslel sa. Ak ideme viac k severu, to je vlastne medzi severom a severovýchodom. Ak je to bližšie k tomu severovýchodu, tak je to medzi severovýchodom a severoseverovýchodom a presne uprostred medzi nimi bol smer, ktorým išli. Kubko teda spokojne vyhlásil: “Ideme presne na severovýchodoseveroseverovýchod.” Maťko sa usmial, a povedal: „Ďakujem, teraz viem, že ideme správne.“

Viete aj vy akým smerom idú dvojčičky Maťko a Kubko?

## Úloha

Na vstupe dostanete názov nejakej svetovej strany a vašou úlohou je zistiť, akým smerom sa nachádza táto svetová strana. Aby sme vám to zjednodušili<sup>4</sup>, chceme od vás uhol medzi severom a danou svetovou stranou, pri čom nechceme radiány, stupne, ani žiadne iné exotické zápisy, ale jednoducho číslo od 0 do 1. A toto číslo chceme v dvojkovej sústave.

Názov svetovej strany vždy vznikne spojením dvoch svetových strán medzi ktorými sa nachádza. Nie však hocikajkých (juho-severovýchod predsa neexistuje). Musia to byť strany “o úroveň vyššie”. Predstavme si že na prvej úrovni je sever, východ, juh, západ. Na druhej pridáme **aj** tie medzi nimi (teda S,SV,V,JV,J,JZ,Z,SZ). Môžete si všimnúť, že každá ďalšia nová strana už bude vznikáť medzi dvoma stranami, z ktorých jedna vznikla na vyššej úrovni ako druhá. V takomto poradí ich aj skladáme. Máme teda severo-severovýchod a nie severovýchodo-sever, lebo sever vznikol už na prvej úrovni a severovýchod až na druhej.

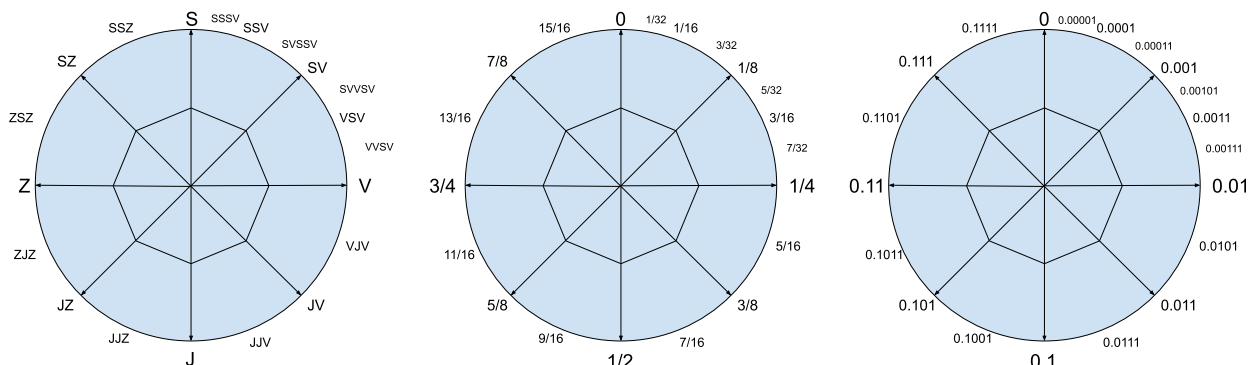
Teraz sa pozrieme na uhly. Sever je 0, východ je  $1/4$ , juh je  $1/2$  a západ  $3/4$ . Uhol 1 by opäť predstavoval sever, ale to nebudeme používať, lebo sever je už 0. Ďalšie uhly sú jednoducho uprostred medzi susedmi. Juhozápad je teda medzi  $1/2$  a  $3/4$ , a má uhol  $5/8$ .

<sup>4</sup>Áno, naozaj je to jednoduchšie.

Na koniec sa pozrieme ako vyzerá **binárny zápis**<sup>5</sup>. Nula je stále nula. V binárnej sústave predstavujú miesta naľavo smerom od desatinnej čiarky postupne hodnoty 1, 2, 4, 8, 16, ... a tie na pravej strane, (desatinné<sup>6</sup> miesta) predstavujú zase postupne z ľava do prava 1/2, 1/4, 1/8, 1/16, ... .

|                     |   |   |   |   |   |   |    |    |   |                                   |
|---------------------|---|---|---|---|---|---|----|----|---|-----------------------------------|
| 1 0 1 1 . 0 1 1 0 1 |   |   |   |   |   |   |    |    |   |                                   |
| 8                   | 4 | 2 | 1 | 1 | 1 | 1 | 1  | 1  | 1 | = 8+2+1+1/4+1/8+1/32 = 11 + 13/32 |
|                     |   |   |   | 1 | 1 | 1 | 1  | 1  | 1 |                                   |
|                     |   |   |   | 2 | 4 | 8 | 16 | 32 |   |                                   |

Napríklad  $13/16 = 1/2 + 1/4 + 0/8 + 1/16$ , teda 0.1101 v dvojkovej sústave<sup>7</sup>.



### Formát vstupu

Na vstupe je jeden riadok a na ňom skratka názvu svetovej strany. Skladá sa z veľkých písmen "S,V,Z,J". Dĺžka tohoto slova nepresiahne 10<sup>7</sup>

### Formát výstupu

Vypíšte jedno desatinné číslo v dvojkovej sústave, bez núl na konci s výnimkou nuly samotnej pre sever. Na oddelenie desatinných miest použite bodku, nie čiarku.

### Príklady

|       |        |
|-------|--------|
| vstup | výstup |
| SV    | 0.001  |

*Sevorovýchod,  $1/8 = 0/2 + 0/4 + 1/8$*

|       |         |
|-------|---------|
| vstup | výstup  |
| SVVSV | 0.00101 |

*Sevorovýchodo-Východosevorovýchod,  $5/32 = 0/2 + 0/4 + 1/8 + 0/16 + 1/32$*

## 5. Interpunkčné párovanie

12 b za popis, 8 b za program

Igor a Ignác sú, ako inak, dvojčičky. Nepoznajú nudu, keďže odmalička sa vo všetkom predbiehajú. Ten začal prvý rozprávať, tamten prvý napočítal do desať, jeden zahral na klavíri Ódu na radosť, druhý nakreslil slušnú trojbodovú perspektívu, atď. Ešte aj v škole sa im zapáčilo rovnaké dievča. Renáta mala ale oči pre iného a bratia sa tak dlho na seba nehnevali.

"Počuj, Ignác," začal včera jeden z bratov. "Ukážem ti zátvorkovú šifru, ktorú som dnes vymyslel. Napíšem nejaký zátvorkový reťazec z (, ), [, ], { a }, v ktorom práve jedna zátvorka nemá pár."

"Krása..." odvetil Ignác.

<sup>5</sup>[https://en.wikipedia.org/wiki/Binary\\_number#Decimal](https://en.wikipedia.org/wiki/Binary_number#Decimal)

<sup>6</sup>v dvojkovej sústave skôr polovičné

<sup>7</sup>pre odvážnejších: zapíšte v dvojkovej sústave 1/3

“Čakaj, to ešte nie je všetko. Nemám to úplne domyslené, ale ak je možné chýbajúcu zátvorku do reťazcu doplniť tak, aby bol správne ozátvorkovaný, hodnota správy je 1, inak je to 0,” objasnil Igor.

“Eh, je pravda, že pomocou 1 a 0 by sme vedeli po bitoch posielat ľubovoľné správy, ale bolo by to dosť nepraktické. Pravdaže, ak by sme dokázali v jednom reťazci uchovať väčšie číslo, ktorým by sa dala reprezentovať komplexnejšia informácia, dosť by sa používanie tejto šifry zjednodušilo,” navrhuje Ignác.

“Ako by sa to dalo spraviť?”

“Rovnako, ako si to vymyslel, ale trochu zložitejšie. Namiesto toho, či sa chýbajúca zátvorka dá vložiť do reťazca, spočítame, na koľko rôznych pozícií sa dá vložiť, pričom vždy bude možné zátvorku správne vložiť.”

“Výborne! A keďže zátvorky sú párové interpunkčné znamienka, dáme šifre aj taký názov.”

“No dobre.”

I keď chlapi už majú program, ktorý dokáže zakódovať nejakú správu do Interpunkčného párovania, nezvýšil sa im už žiaden čas, aby vytvorili dešifrovací program. Ešteže existujú šikovní KSPáci, ktorí im s tým pomôžu.

## Úloha

Na vstupe je reťazec zo zátvoriek (, ), [, ], { a }. Tento reťazec bol najprv správne ozátvorkovaný, no práve jedna zátvorka z neho zmizla. Zistite, na koľko rôznych miest môžeme chýbajúcu zátvorku doplniť tak, aby bolo ozátvorkovanie korektné. Napríklad [{}] je nesprávne ozátvorkovanie. Tak isto aj ){}( alebo ({}). Naopak ({}), {{{}}(){} alebo []{}() sú správne ozátvorkovania.

### Formát vstupu

V prvom riadku vstupu je číslo  $1 \leq t \leq 1000$  udávajúce počet reťazcov. Nasleduje  $t$  riadkov. V každom riadku sa nachádza reťazec  $n$  znakov zátvoriek (, ), [, ], { a }. Práve jedna zátvorka v reťazci chýba. Platí:  $n = k + 1$  a  $0 \leq k \leq 2000$ .

### Formát výstupu

Vypíšte jedno celé číslo: počet miest v reťazci, kam vieme korektné vložiť chýbajúcu zátvorku.

### Príklady

| vstup                                | výstup           |
|--------------------------------------|------------------|
| 4<br>{[{}]<br>[(){}<br>[[([()]]<br>[ | 1<br>3<br>6<br>1 |

## 6. Čakove deti

12 b za popis, 8 b za program

Ako všetci vieme tak Čak Norris má dvojičky - Čaka a Čakinku. Prázdniny už skončili a obaja tak už musia chodiť do školy. Dokonca už dostali aj prvú domácu úlohu. Majú dokázať, že Zem je plochá. Inými slovami majú ukázať, že to je pravda.

Ich tatko, starý Norris kedysi dávno vytesal všetky všeobecne známe praudy na neďalekú skalú. Čak a Čakinka by boli radi ako on a preto domácu úlohu vypracujú práve na túto skalú.

Čakinka je však lenivá, nechce sa jej chodiť až ku skale. Avšak je aj výnimočne bystrá a má skvelé dedukčné schopnosti. A tak ako správny súrodenci si s touto neľahkou úlohou pomôžu. Ich stratégia je nasledovná.

Čak bude behať ku skale a vytesávať novo zistené praudy. Cestou naspäť si niekoľko z už vytesaných práud zapamätá. Dobehe domov ku Čakinke povie jej čo si pamätá a bystrá Čakinka z toho odvodí neaký ďalší výrok (po tomto odvodení sa už aj tento výrok stáva praudou). Čak následne beží ku skale a aj tam tento výrok vytesá, ak tam ešte nie je. Celý cyklus opakujú kým na skale ešte nie je vytesaná Zem je plochá.

Čak však nemá moc dobrú pamäť vie si pamätať len obmedzený počet výrokov, keď beží naspäť od skaly. Pomôžte im a zistite koľko pamäte Čakovi stačí. Teda zistite koľko výrokov si musí vedieť Čak naraz pamätať (na jednej ceste od skaly domov) aby boli schopný splniť domácu úlohu alebo vypíšte, že to nie je možné bez ohľadu na jeho pamäť.

### Úloha

Vo všeobecnosti existuje  $n$  výrokov, ktoré si očísľujeme od 0 po  $n - 1$ . Výrok číslo 0 je “Zem je plochá” a tento dostali za domácu úlohu Čak a Čakinka dokázať.

Ďalej máme  $m$  odvodzovacích pravidiel, pričom  $i$ -te z nich pozná Ćakinka v tvare  $a_{i,0}, \dots, a_{i,k_i}$ . Ćo znamená, že ak sú výroky  $a_{i,0}, \dots, a_{i,k_{i-1}}$  praua, tak aj výrok  $a_{i,k_i}$  je praua.

A na záver máme na skale vytesaných  $l$  výrokov, ktoré sú všeobecná praua (netreba ich odvádzať).

Zistite najmenší počet výrokov, ktoré si musí Ćak pamätať naraz počas cesty domov, pokiaľ Ćakinka vie odvádzať len z výrokov, ktoré si Ćak pamätá. Alebo vypíšte, že by nám nestačila žiadna pamäť.

### Formát vstupu

V prvom riadku vstupu sú čísla  $n$  a  $m$  ( $1 \leq m, n \leq 1\,100\,000$ ) udávajúce počet možných výrokov a počet odvodzovacích pravidiel.

Nasleduje  $2m$  riadkov. V nepárnych riadkoch je počet výrokov  $k_i$  v odvodzovacej formuly, a v párnych samotné čísla výrokov v odvodzovacom pravidle  $a_0, \dots, a_{k,i}$ . Pritom môžete predpokladať, že  $\sum_{i=0}^{m-1} k_i \leq 2\,000\,000$ .

Na záver je vo vstupe počet všeobecných práud  $1 \leq l \leq n$ . V poslednom riadku sú ich čísla.

### Formát výstupu

Na výstup vypíšte jediné číslo a to najmenšiu možnú pamäť, pre ktorú vieme dokázať výrok “Zem je plochá”. Ak to nie je možné pre žiadnu pamäť vypíšte  $-1$ .

### Príklady

| vstup                              | výstup |
|------------------------------------|--------|
| <pre>3 2 1 1 0 2 1 2 0 2 1 2</pre> | 1      |

*Zjavne si stačí zapamätať všeobecnú prauu 1 a potom už len použijeme prvé odvodzovacie pravidlo*

| vstup                                      | výstup |
|--|--------|
| <pre>4 3 2 1 2 3 2 1 3 2 2 2 3 0 1 1</pre> | -1     |

*Na dokázanie plochosti zeme potrebujeme najprv dokázať prauu 2 a 3. Na dokázanie prauy 2 však potrebujeme prauu 3, a na dokázanie prauy 3 potrebujeme zasa prauu 2. V tejto čudesnej, fiktívnej realite teda zem plochá asi nieje.*

## 7. Keď už je to po ceste...

12 b za popis, 8 b za program

V Manhattane kedysi stáli dvojččky. Kolumbus objavil Ameriku, one thing led to another a dvojččky dnes už v Manhattane nestoja. Finančníci, ktorí v dvojččkách pracovali, sa rozprchli po celom Manhattane.

Nieje nad vzťah založený na peniazoch, a tak ešte dodnes neustále medzi sebou kadečo vybavujú – odcestujú na nové pracovisko ich známeho, prehrabnú sa v kufríku plnom papierov, podpíšu niečo čo nikto nikdy nečítal, a vrátia sa naspäť do práce. A keď sa vydarí, po ceste sa ešte zastavia u kamaráta, vypijú kávu, zaspomínajú na strašidelné časy kedy si aj úplne obyčajní ľudia vedeli dovoliť nový, priestorný byt, a pokračujú v pracovnej ceste. Svojim nadriadeným toto drobné zdržanie vysvetlia ako zápchu v poludňajšej New Yorkskej<sup>8</sup> premávke, a veselo si celú cestu pripíšu do firemného času.

Ale takéto výhovorky nefungujú vždy – ak by kvôli priateľskej návšteve finančník robil dajaké obkľuky, určite by mu na to nadriadení prišli<sup>9</sup>. Takúto zastávku sa teda odhodlajú urobiť len vtedy, keď nezvýšia prejdenú vzdialenosť k ich ozajstnej destinácii.

<sup>8</sup>Kedy ste naposledy videli v slove ‘rksk’?

<sup>9</sup>pomocou GPS trackingu firemných vozidiel



FKS<sup>10</sup> sa rozhodol vyčíslieť, koľko takýchto návštev môže denne prebiehať. V rámci znižovania nákladov však programátorské práce outsourceli na Slovenské stredné školy...

## Úloha

Všetky budovy Manhattanu perfektne zapadajú do štvorcovej mriežky.

Zadané sú súradnice budov v Manhattane, v ktorých pracujú finančníci.

Keďže sú v Manhattane, vzdialenosť medzi budovami je samozrejme Manhattanovská – budova so súradnicami  $(x_1, y_1)$  je od budovy so súradnicami  $(x_2, y_2)$  vzdialená  $|x_1 - x_2| + |y_1 - y_2|$ .

Finančník cestujúc z budovy A do budovy B sa môže po ceste zastaviť pri budove C práve vtedy, keď vzdialenosť budov A a B je rovná vzdialenosti budov A a C plus vzdialenosti budov C a B.

Vyhodnotte, koľko rôznych počestných návštev sa môže v Manhattane odohrať. Dve návštevy považujeme za rôzne ak sa líšia v štartovnej, cieľovej, alebo navštívenej budove.

## Formát vstupu

V prvom riadku je číslo  $n$  – počet finančníkov v Manhattane. Každý z nasledujúcich  $n$  riadkov obsahuje dve celé čísla  $x_i, y_i$  – súradnice budovy, v ktorej pracuje  $i$ -ty finančník. Súradnice budov sú navzájom rôzne.

Pre testovacie sady pritom platí nasledovné:

| Sada                   | 1     | 2     | 3       | 4         |
|------------------------|-------|-------|---------|-----------|
| $1 \leq n \leq$        | 200   | 2 000 | 200 000 | 200 000   |
| $0 \leq x_i, y_i \leq$ | 1 000 | 1 000 | 1 000   | 1 000 000 |

## Formát výstupu

Vypíšte jediné číslo – počet rôznych návštev, ktoré medzi sebou dokážu finančníci urobiť.

## Príklady

vstup

```
3
2 2
3 2
4 2
```

výstup

```
2
```

*Návštevu v druhej budove si môžu dovoliť finančníci cestujúci z prvej do tretej budovy, a naopak.*

vstup

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

výstup

```
8
```

*Teraz môžu navyše prvú budovu navštíviť finančníci zo štvrtej budovy na ceste do druhej alebo tretej (a naopak), a druhú budovu na ceste do tretej (a naopak).*

## 8. Yvettin Ementál Neschopnosti

12 b za popis, 8 b za program

Yvette sa na klávesnici pokazil medzerník. Hneď o tom aj napísala svojej kamarátke Xénii. Samozrejme, v rámci dobrých zvykov, bez diakritiky, interpunkcie a veľkých písmen:

ahojxenianeuverisalepokazilsamimedzernik

Lúštenie tejto správy Xénii chvíľku trvalo, no celkom sa jej to zapáčilo. Odpísala teda:

jeeejtojesrandapodmesipisatbezmedzier

Yvette lúštenie správy spočiatku veľmi nešlo. “To je nad moje mentálne schopnosti,” pomyslela si. Nakoniec však správu rozlúštila, a keďže nechcela Xéniu sklamať, odpovedala:

<sup>10</sup>Finačný Korupčný Sektor

Táto správa Xéniu úprimne zmiatla. “Nad môj Ementál Neschopnosti? Čo to je? Už som počula o Prsteni Neviditeľnosti, Opasku Sily, aj Papučiach Obratnosti, ale o Ementáli Neschopnosti som ešte nepočula.”

Po chvíli zmätku a dvoch telefonátoch dievčatá pochopili, že ak si budú písať bez medzier, občas sa nejaká správa bude dať rozdeliť na slová viacerými spôsobmi. Preto sa rozhodli, že pri písaní budú používať iba obmedzenú množinu slov, tak, aby sa im toto nemohlo stať. Ako však zistíš, či je nejaká množina slov dobrá?

## Úloha

Majme zoznam  $k$  rôznych slov. Dve rôzne postupnosti slov zo zoznamu budeme volať *dvojičky*, ak zreťazením slov z jednej postupnosti vznikne rovnaký reťazec ako zreťazením slov druhej postupnosti. Napríklad, ak by náš zoznam obsahoval slová {emental, mentalne, moj, moje, neschopnosti, schopnosti}, postupnosti (moje, mentalne, schopnosti) a (moj, emental, neschopnosti) sú dvojičky. Slová sa môžu v rámci postupností aj opakovať, postupnosti však musia byť rôzne. Teda napríklad postupnosť (moje, mentalne, schopnosti, moje, moje) je dvojička s postupnosťou (moj, emental, neschopnosti, moje, moje), ale postupnosti (moj, moje) a (moj, moje) dvojičky nie sú, lebo nie sú rôzne.

Vašou úlohou bude pre zadaný zoznam slov rozhodnúť, či preň existujú nejaké dvojičky.

## Formát vstupu

Aby sme vám sťažili náhodné hádanie správnych odpovedí, v jednom vstupnom súbore bude niekoľko zadaní úloh.

V prvom riadku bude jedno celé číslo  $t$  ( $1 \leq t \leq 10$ ) – počet zadaní, ktoré máte vyriešiť. Ďalej je na vstupe  $t$  zadaní, každé z nich má nasledujúci formát.

V prvom riadku zadania je jedno celé číslo  $k$  ( $1 \leq k \leq 200,000$ ) – počet slov v zozname. Nasleduje  $k$  riadkov, každý z nich obsahuje jeden neprázdny reťazec tvorený malými písmenami anglickej abecedy (a-z) – jedno slovo zo zoznamu. Všetky tieto slová sú navzájom rôzne.

Súčet dĺžok všetkých slov vo vstupnom súbore (teda zo všetkých  $t$  zadaní dokopy) označme  $N$ . Bude platiť  $N \leq 200,000$ .

## Formát výstupu

Pre každé zadanie zo vstupu vypíšte (v rovnakom poradí) jeden riadok. Ak pre zadaný zoznam slov existujú dvojičky, riadok nech obsahuje iba slovo **ano**, ak neexistujú, riadok nech obsahuje iba slovo **nie**.

## Hodnotenie

Vaše programy budeme testovať na štyroch skupinách testovacích vstupov. Pre jednotlivé skupiny budú platiť nasledujúce obmedzenia:

| Skupina         | 1   | 2     | 3       | 4       |
|-----------------|-----|-------|---------|---------|
| $1 \leq N \leq$ | 100 | 2 000 | 200 000 | 200 000 |
| $1 \leq k \leq$ | 20  | 100   | 100     | 200 000 |

Súčet dĺžok všetkých slov v **jednom** zadaní označme  $n$ . Vo vašich riešeniach uvádzajte časovú zložitosť vášho algoritmu v závislosti od parametrov  $n$  (celková dĺžka slov v zozname) a  $k$  (počet slov v zozname), teda nie od parametrov  $t$  a  $N$ .

Okrem toho uveďte aj odhad časovej zložitosti vášho algoritmu iba v závislosti od parametra  $n$ . Ak by teda napríklad váš algoritmus bežal v čase  $O(n^2k^3)$ , môžete uviesť, že beží v čase  $O(n^5)$  (keďže  $k \leq n$  na každom vstupe).

## Príklad

vstup

```
3
6
emental
mentalne
moj
moje
neschopnosti
schopnosti
4
korespondencny
seminar
z
programovania
4
aaa
aaab
bc
caaa
```

výstup

```
ano
nie
ano
```

*Najkratšie dvojčky v poslednom zadaní sú postupnosti (aaab, caaa) a (aaa, bc, aaa).*