



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. :)
- **Odobzdaj** riešenia úloh. Odkaz na odovzdávanie úloh nájdeš pod webovým zadaním každej úlohy alebo na stránke <https://www.ksp.sk/odovzdavanie>. Na odovzdávanie sa treba prihlásiť, aby sme vedeli, komu máme dať body.
 - Vo väčšine úloh odovzdávaš program a popis.
 - Program je hneď po odovzdaní otestovaný testovačom a hneď vidíš, koľko bodov za program máš. Program môžeš odovzdávať znova a znova, až kým nie si spokojný s výsledkom. Ak nevieš, ako majú vyzeráť odovzdané programy, pozri si <https://www.ksp.sk/odovzdavanie-programov>
 - Do popisu slovne napíšeš, ako tvoje riešenie funguje, prečo funguje a tiež odhad časovej a pamätovej zložitosti programu. Viac sa dozvieš na stránke <https://www.ksp.sk/ako-riesit>. Popis opraví a obodujú vedúci KSP po skončení kola.
- Po skončení kola si **prečítaj vzorové riešenia** úloh (veľa sa z toho naučíš), pozri svoje opravené popisy (či ti tam vedúci nenapísali nejaké poučné komentáre), pozri sa do výsledkovky a **teš sa**, koľko máš bodov. Vo výsledkoch sa hodnotí samostatne letná a zimná časť. V každej časti je dôležitý celkový súčet bodov.
- Prečo sa máš tešiť z bodov? Čítaj ďalej.

Čo môžem vyhrať?

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

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

Pravidlá a levely

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

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

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

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

Registrácia

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

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



Úlohy 1. kola letnej časti

Termín odoslania riešení tohto kola je **17. marca 2025**. Doprogramovanie končí 31. marca 2025.

1. Malá mäťúca prechádzka

12 b za popis, 8 b za program

V jednej malej dedinke, nazývanej Štvorečkovo, žila mačka Micka s jej mačiatkom Cickou. Cicka ešte nevie, ako Štvorečkovo vyzerá, tak Micka s ňou chodí každý deň na krátku obchádzku po dedinke. Ľudia v Štvorečkove bývajú v kockatých domoch so štvorčekovým pozemkom a Micka ich rada navštevuje. Micka s Cickou rady chodia po susedoch a stretávajú známe tváre. Vždy sa ale vrátia domov, kde si po prechádzke dajú poriadnu siestu. Raz sa Micka dala viesť, aby vyskúšala, ako sa Cicka naučila chodiť po dedine. Po tejto prechádzke chce Micka povedať Cicke koľko krokov spravila zle, ale nevie koľko. Pomôž Micke nájsť, koľko krokov spravila Cicka zle.

Úloha

Máme nekonečnú štvorčekovú mapu a vieme sa pohybovať po nej zvislo a vodorovne. Dostaneme záznam cesty, ktorou Cicka šla po dedinke. Jeden znak reprezentuje posun na mape o jeden štvoreček. Cicka sa chcela vrátiť do bodu, v ktorom začínala, ale niekoľkokrát sa pomýlila. Zistíte, koľko najmenejkrát sa mohla Cicka na svojej prechádzke pomýliť. Môžete predpokladať, že Cicka spravila párny počet krokov.

Formát vstupu

V prvom riadku vstupu je číslo n ($1 \leq n \leq 10^6$) dĺžku cesty mačiatka. V druhom riadku vstupu je popis cesty - reťazec dĺžky n skladajúci sa so znakov L, P, H a D (dolava, doprava, hore a dole).

Pre jednotlivé sady platia takéto obmedzenia:

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

Formát výstupu

Vypíš jeden riadok a v ňom jedno celé číslo, ktoré znázorňuje najmenší počet chýb v prechádzke.

Príklady

vstup	výstup
6 DDLLHH	1

Pre tento prípad, ak by sme chceli sa vrátiť na pôvodné miesto, tak namiesto jedného kroku dolava by sme mali ísť doprava.

vstup	výstup
8 DDLLHHPP	0

Urobili sme jeden okruh a vrátili sme sa naspäť, odkiaľ sme začali.

2. Ako to len ofarbiť

12 b za popis, 8 b za program

Miška veľmi rada vymýšľa. Prednedávnom si chcela zaobstarať novú malú mačku a jej priateľ musel vynaložiť všetky svoje sily na to, aby ju prehovoril, že je to hlúposť. Teraz si vymyslela, že sa jej nepáči, ako vyzerá jej terasa.

Miškina terasa sa skladá z $N \times M$ rovnako veľkých štvorcových kachličiek, ktoré tvoria obdĺžnik. Aktuálne majú všetky kachličky jednu z dvoch farieb - ružovú alebo fialovú. Miška je s farbami spokojná, chce však, aby jej terasa vyzerala trochu inak. Chce, aby sa žiadne dve kachličky s rovnakou farbou nedotýkali hranou (rohom

môžu). Keďže hádať sa s Miškou je ťažšie než jednoducho premaľovať terasu, Miškin priateľ by rád vedel, koľko najmenej kachličiek treba premaľovať, aby bola Miška spokojná.

Formát vstupu

V prvom riadku vstupu dostanete dve čísla n ($1 \leq n \leq 10^6$) - výšku terasy, a m ($1 \leq m \leq 10^6$) - šírku terasy.

Nasleduje n riadkov. V každom riadku je reťazec dĺžky m pozostávajúci z písmeniek r (ružová kachlička) alebo f (fialová kachlička).

V sadách platia nasledujúce obmedzenia:

Sada	1	2	3	4
$1 \leq N \times M \leq$	20	10^3	10^5	10^6

Formát výstupu

Na jeden riadok vypíšete jedno číslo - koľko najmenej kachličiek musíme premaľovať tak, aby sa žiadne dve kachličky rovnakej farby nedotýkali hranou.

Príklad

vstup	výstup
<pre>1 1 r</pre>	<pre>0</pre>

Tu netreba prefarbovať žiadnu kachličku.

vstup	výstup
<pre>2 3 rfr frf</pre>	<pre>0</pre>

Taktiež nemusíme nič prefarbovať.

vstup	výstup
<pre>2 2 rf rr</pre>	<pre>1</pre>

Potrebuje prefarbiť ľavé dolné políčko na fialové, čo je jedno prefarbenie.

3. Častá naháňačka

12 b za popis, 8 b za program

MisQo sa veľmi rád pozerá na hru Subway Surfers. Tak si raz povedal, že si ju skúsi naprogramovať. Keďže nie je moc dobrý programátor, rozhodol sa hru nasledovne zjednodušiť. Keďže má veľmi rád mačky, rozhodol sa, že hráč bude myš a mačka ju bude naháňať, bude sa mu na to potom ešte lepšie pozeráť.

Úloha

MisQova verzia Subway Surfers pozostáva z 2 vertikálnych línií, na ktorých sa dá hýbať. Podobne ako v originálnej hre tu musíme preskakovať prekážky. Tu ale hráme za myš a naháňa nás agresívna mačka. Myš môže robiť iba nasledovné 3 akcie:

- posunúť sa o 1 dopredu.
- posunúť sa o 1 dozadu.
- skočiť - zmení sa línia v ktorej sa myš nachádza a zároveň sa posunie o k dopredu.

Po každom pohybe hráča sa mačka posunie o 1 dopredu. Ak nastane situácia, že sa myš a mačka nachádzajú na rovnakej vzdialenosti od začiatku (Mačka je veľká, teda zaberá obe línie naraz), prípadne mačka prebehne myš, mačka myš zje a tým pádom myš prehrala.

Na rozdiel od originálnej hry, táto verzia má koniec, keď sa dostaneš na koniec levelu. Rozhodni, či pre dané prekážky sa dá level vyhrať. To znamená, že sa dá dostať aspoň na koniec levelu bez toho, aby ťa mačka zjedla. (koniec vieme aj preskočiť - ak má level 15 políčok a my skočíme rovno na 18., tak sme vyhrali).

Na začiatku hry sa myš nachádza na 1. pozícii v 1. línii a mačka na 0. pozícii tesne za ňou. Je garantované, že na tejto pozícii nie je prekážka.

Formát vstupu

V prvom riadku vstupu je číslo n ($1 \leq n \leq 10^6$) – dĺžka levelu, a číslo k ($1 \leq k \leq n$) – dĺžka myšieho kroku.

V druhom riadku je n znakov - mapa prvej línie hry, kde znak X značí prekážku a znak = značí voľno. V treťom riadku je znova n znakov - mapa druhej línie hry, kde znak X značí prekážku a znak = značí voľno.

Úloha má osem sád vstupov, v ktorých platia nasledovné obmedzenia:

Sada	1, 2	3, 4	5, 6	7, 8
$1 \leq n \leq$	20	100	1 000	10^5

Formát výstupu

Vypíš jeden riadok a v ňom **Vyhral som**, ak sa dá level vyhrať, inak **Prehral som**.

Príklady

vstup	výstup
<pre>8 4 ==XX=XX= ==X==X==</pre>	<pre>Vyhral som</pre>

Myši stačí iba 2-krát skočiť - po prvom skoku budeme na 5-tom políčku v druhej línii, odkiaľ už vieme rovno skočiť do cieľa. Rovnako by sme mohli po prvom skoku ísť dozadu a potom skočiť

vstup	výstup
<pre>8 3 =X=XXXX= =X==XX==</pre>	<pre>Prehral som</pre>

Ako prvý krok musí myš skočiť, lebo hneď pred nami je prekážka. Po skoku sa dostaneme na 4. políčko v druhej línii. Jediný možný tah je ísť dozadu. Potom jediný možný tah je ísť naspäť dopredu. Takto sa budeme cykliť až kým nás mačka nechytí, teda prehráme.

4. Iné jazýčky

12 b za popis, 8 b za program

V reakcii na [vedeckú štúdiu](#)¹ publikovanú v časopise sa profesor Labka rozhodol, že všetkým raz a navždy dokáže svoju dlhoročnú teóriu o možnostiach komunikácie medzi ľuďmi a mačkami.

Jeho teória je vskutku priamočiara – mačky hovoria tie isté slová, čo my (pretože sa ich všetky učia od svojich ľudských majiteľov), akurát nepoznajú naše hlásky, takže používajú iné, svoje. Takže vraj stačí zistiť, ktoré ľudské hlásky sa prekladajú na ktoré mačacie a naopak, a môžeme plne chápať mačaciu reč...

Vy samozrejme viete, že mačky podstatné informácie komunikujú výhradne telepaticky, takže jeho tvrdeniam neveríte.

Vaším cieľom je preto jeho hypotézu vyvrátiť, no musíte to spraviť veľmi opatrne, aby sa na vás akademická obec nepozerala cez ~~labky~~prsty (nájsť si takú metriku, ktorá by jeho tvrdeniam nedala šancu, by totiž nebolo dostatočne vedecké). Keďže profesor Labka pripúšťa rozdielny slovosled ľudskej a mačacej reči, musíte počítať aj s možnosťou, že preložiť sa občas dajú iba časti viet, nie vety celé. Samozrejme, ak takých častí bude príliš málo, alebo budú veľmi krátke, je jasné, že ide o štatisticky zanedbateľnú náhodu a jeho teória je úplný nezmysel.

Úloha

Dostanete dva reťazce dĺžky n zložené z písmen anglickej abecedy – vetu L povedanú človekom (veľkými písmenami) a vetu M (malými písmenami), o ktorej profesor Labka tvrdí, že je jej mačacím ekvivalentom.

Vaším cieľom je nájsť všetky preložiteľné intervaly – teda dvojice $1 \leq i \leq j \leq n$ začiatku a konca podreťazcov – také, že podreťazec $L[i, j]$ v ľudskom reťazci sa dá preložiť na podreťazec $M[i, j]$ v mačacom reťazci.

Podreťazec sa dá preložiť práve vtedy, keď existuje nesporné kódovanie písmen z ľudského reťazca do mačacieho a naopak – teda keď platí: pokiaľ sú v jednom z podreťazcov dve rovnaké písmená, budú rovnaké aj písmená na zodpovedajúcich pozíciách v druhom podreťazci.

¹<https://www.sciencedirect.com/science/article/pii/S016815912300309X>

Keďže dlhšie podreťazce prinášajú viac dôkazov o (ne)správnosti hypotézy, považujeme ich za dôležitejší dôkazový materiál – pre každý preložiteľný interval teda zistíte jeho dĺžku a odpovedzte ich celkový súčet.

Formát vstupu

V prvom riadku vstupu je číslo n ($1 \leq n \leq 10^6$) – dĺžka oboch reťazcov.

Na ďalších dvoch riadkoch sú reťazce L a M , vždy v tomto poradí.

V jednotlivých sadách platia nasledovné obmedzenia:

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

Formát výstupu

Vypíšte jeden riadok a v ňom jedno celé číslo – súčet dĺžok všetkých preložiteľných intervalov pre danú dvojicu reťazcov L a M .

Príklady

vstup

```
3
AAB
baa
```

výstup

```
3
```

Preložiteľné sú len intervaly $[1-1]$, $[2-2]$, $[3-3]$, keďže v intervaloch $[1-2]$ a $[2-3]$ kolidujú postupne preklady písmen A a a .

vstup

```
6
AHOJKY
mnauky
```

výstup

```
56
```

Žiadny z podreťazcov (ktorých je dokopy 21) nie je nepreložiteľný (nie je to prekvapivé, keďže v žiadnom z nich sa neopakujú písmená). Máme teda 1 interval dĺžky 6, 2 intervaly dĺžky 5... až po 6 intervalov dĺžky 1.

vstup

```
8
AHOJACAU
mnaumnau
```

výstup

```
54
```

5. A koľko mu mám dať?

12 b za popis, 8 b za program

Martin má kocúra Ferka. Ferko je ale mierne obézny kocúr, takže musí začať držať diétu. Martin počul o jednej veľmi efektívnej diéte pre kocúrov - pórová diéta. V tejto diéte môžu mačky jesť iba pór.

Bohužiaľ prestup na túto diétu Ferkovi nepomohol. To preto, že mu Martin dával príliš veľa póru. Na internetovom fóre Krímenie Super Partákov sa dozvedel, že na každej škatuli od póru sú čísla, ktoré určujú, koľko póru má dať svojmu miláčikovi. No a keďže je to pór, tak sa to zisťuje pOR-om. Pomôžete Martinovi zistiť, koľko póru má dať Ferkovi?

Úloha

Máme pole čísiel P . Zaujíma nás, koľko je v ňom takých súvislých úsekov, ktoré majú bitový OR rovný k .

Formát vstupu

V prvom riadku vstupu dostanete dve čísla - n - počet prvkov v poli P a k - bitový OR ktorý sa snažíme docieľiť.

Ďalej bude nasledovať jeden riadok, na ktorom bude n nezáporných celých čísiel.

V jednotlivých sadách platia nasledujúce obmedzenia:

Sada	1	2	3	4
$1 \leq n \leq$	100	10 000	10^5	$5 \cdot 10^5$
$1 \leq k \leq$	1 000	10^6	10^9	10^9
$1 \leq P_i \leq$	1 000	10^6	10^9	10^9

Formát výstupu

Vypíšte počet úsekov poľa P , ktoré majú bitový OR rovný k

Príklady

vstup	výstup
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> 2 1 1 1 </div>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> 3 </div>

Úseky, ktoré majú bitový OR 1 sú: $[0, 0]$, $[0, 1]$, $[1, 1]$

vstup	výstup
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> 10 1 1 1 1 1 1 1 1 1 1 1 </div>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> 55 </div>

Každý súvislý podúsek má bitový OR 1 (podúsekov je 55)

vstup	výstup
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> 5 7 1 2 2 1 1 </div>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> 0 </div>

V žiadnom podúseku nie je bitový OR 7

6. TokTik Mačiatka

12 b za popis, 8 b za program

Mňau! Alicka má rada pozeranie smiešnych videí s mačičkami² na jej obľúbenej platforme na väznenie používateľov - TokTiku. Je to jej jediné potešenie v celom živote, pretože predsa čo iné okrem sledovania rozkošných (a občas explodujúcich) mňaukadiel by v cez deň robila? Toto však má aj neželané následky. A to také, že keďže Alicka týchto videí pozerá veľa, už skoro všetky pozná! Veľmi ľahko sa tak začne nudiť, hlavne ak jej platforma naservíruje video podobné nejakému, ktoré už v ten deň videla. Algorimus platformy však programujú veľmi milí ľudia, ktorí chcú Alicke iba to najlepšie - a teda, aby bola z videí čo najviac šťastná. Rôzne videá totiž Alicku obšťastnia rôzne, napríklad video kde mačička robí salto ju poteší menej ako video, kde nejaké chutnučké mačiatko mňauká. Títo programátori sú však veľmi neschopní, a tak im musíte pomôcť.

Úloha

Na platforme je n nových mačičkových videí, ktoré Alicka ešte nevidela. Tieto videá Alicku potešia rôzne - každé video má svoju kvalitu a_i - teda koľko šťastia toto video dá Alicke po tom, čo si ho pozrie. Niektoré videá sú však podobné niektorým iným videám, pričom pre každé dve videá platí, že existuje práve jedna postupnosť začínajúca jedným a končiaci druhým taká, že každé video je podobné tomu predošlému. Alicka sa začne nudiť (a prestať pozerat videá) práve vtedy, keď začne pozerat nejaké video, ktoré už dnes videla, alebo je podobné nejakému, ktoré dnes už videla. Zistíte, koľko najviac šťastia môže Alicka získať pozeraním týchto videí.

Formát vstupu

V prvom riadku vstupu je číslo n ($1 \leq n \leq 10^6$) udávajúce počet videí.

Druhý riadok obsahuje n medzerou oddelených čísel. i -té z nich udáva hodnotu a_i , teda koľko šťastia dá Alicke i -té video. Nasleduje $n - 1$ riadkov. Na každom riadku sa nachádzajú dve čísla a, b , pričom toto znamená, že video a je podobné videu b (podobnosť je vzájomná, teda aj video b je podobné videu a).

V jednotlivých sadách platia nasledujúce obmedzenia:

Sada	1	2	3	4
$1 \leq N \leq$	20	10^6	10^6	10^6
$1 \leq \max a_i \leq$	10^{12}	10^{12}	10^{12}	10^{12}

²Napríklad [takéto](#).³

Za každú sadu viete získať 2 body. V druhej sade navyše platí, že každé video je podobné najviac dvom iným videám. V tretej sade platí, že každé dve videá dajú Alicke rovnako šťastia, teda $a_0 = a_1 = a_2 \cdots = a_{n-1}$.

Formát výstupu

Vypíš jeden riadok a v ňom jedno celé číslo - najviac šťastia, ako môže Alicka získať.

Príklad

vstup	výstup
<pre>5 15 20 13 24 19 0 3 1 2 2 3 3 4</pre>	<pre>54</pre>

Aj keď by video 3 dalo Alicke najviac šťastia, neoplatí sa jej ho naservírovať. Namiesto toho sa oplatí pozrieť videá 0, 1 a 4.

7. Kocúre a klbko

12 b za popis, 8 b za program

Anička a Bláčik sú dve rozkošné mačiatka, a ako typické mačiatka, ich obľúbená hračka je špagát. Zvykli ho najmä naháňať, ale odkedy zaspali na knižke o teórii hier, osmózou⁴ sa k nim dostali aj nejaké vedomosti, a ich hry začali byť elaborátnejšie.

Konkrétne, keď sa prebudili po výdatnom matematickom spánku, Bláčik a Anička si schmatli dva špagáty, jeden s dĺžkou a a druhý s dĺžkou b mačkometrov, a postupne sa striedajú na ťahu. Mačiatko si vyberie špagát a skrúti ho o celočíselný pozitívny násobok dĺžky druhého špagátu⁵. Samozrejme, mačiatko bez špagátu je smutné mačiatko, a preto ani jedno z nich nechce špagát ťahom úplne zmiznúť.

A keďže mačiatka sú naozaj hyperaktívne, raz ako sa naučili hrať hru, budú ju hrať viackrát. Menovite, klbko z ktorého špagáty berú vystačí na to, aby si vyskúšali hru zo všetkými možnými dĺžkami prvého špagátu v rozmedzí $a_0 \leq a \leq a_1$, a druhého špagátu v rozmedzí $b_0 \leq b \leq b_1$.

Po niekoľkých ďalších šľofíkoch na knihe sa už Anička a Bláčik naučili hrať optimálne a zaujímalo by ich, koľko z týchto hier vyhrá ktoré z nich. Na vylúštenie tejto záhady ešte dost nespali⁶ – hrajú sa predsa so špagátmi! – tak im budete s touto otázkou musieť pomôcť vy!

Úloha

Anička a Bláčik sa hrajú nasledovnú hru. Majú dva špagáty, s dĺžkami a a b mačkometrov. Striedajú sa v ťahoch, mačiatko na ťahu si vždy vyberie celé číslo $k \geq 1$, a buď skrúti prvý špagát o $k \cdot b$ mačkometrov, alebo druhý špagát o $k \cdot a$ mačkometrov. Prehráva mačiatko, ktoré jeden zo špagátov úplne zmizne. Samozrejme, špagát nemožno skrútiť na negatívnu dĺžku.

Hru sa hrajú viackrát, konkrétne pre každú kombináciu a a b v rozsahu $a_0 \leq a \leq a_1$ a $b_0 \leq b \leq b_1$ vrátane (hrajú tak $(a_1 - a_0 + 1)(b_1 - b_0 + 1)$ hier).

Anička vždy začína. Koľko z hier vyhrá, ak obe mačiatka hrajú optimálne?

Formát vstupu

V jednom vstupe sa bude nachádzať viacero testovacích vstupov. Prvý riadok vstupu obsahuje celé číslo $1 \leq t \leq 10^5$ – počet testovacích vstupov.

Na ďalších t riadkoch sa na každom nachádzajú štyri celé čísla oddelené medzerou: a_0, a_1, b_0 a b_1 . Je zaručené, že $1 \leq a_0, a_1, b_0, b_1 \leq 10^9$, $0 \leq b_1 - b_0, a_1 - a_0 \leq 10^6$.

Úloha má štyri sady. V prvej sade platí, že $a_0, a_1, b_0, b_1 \leq 500$. V druhej sade platí, že $a_0 = a_1$ a $b_0 = b_1$. V tretej sade platí, že $a_1 - a_0, b_1 - b_0 \leq 10^5$.

Je garantované, že súčet $a_1 - a_0 + 1$ (resp $b_1 - b_0 + 1$) v prvých troch sadách neprekročí 10^5 , a v štvrtej sade neprekročí $5 \cdot 10^6$.

⁴Osmóza je presun hmoty (v tomto prípade znalostí) cez polopriepustnú membránu (v tomto prípade strany knihy). V prípade, že chcete aj vy skúsiť učenie osmózou, odporúčame zdiemnuť na niektorej zo zbierok KSP.

⁵Odhryzne zvyšok a stratí ho pod gaučom.

⁶Hovorí sa, že mačiatka spia 22 hodín denne, ale povedal to niekto tým mačiatkam?

Formát výstupu

Pre každý z t vstupov vypíšete jeden riadok a v ňom jedno celé číslo: počet hier v ktorých Anička vyhrá, ak obe mačiatka hrajú optimálne.

Dávajte si pozor, že výstup môže byť pomerne *veľký*. Ak programujete v pythone, odporúčame vám vypisovať celý výstup *naraz* pomocou jediného použitia `print`. V prípade, ak používate C++, odporúčame vám používať `\n` namiesto `endl`.

Príklad

vstup	výstup
<pre>2 11 11 2 2 1 6 1 6</pre>	<pre>1 20</pre>

*V prvom prípade hrajú mačiatka jedinou hru: $s a = 11$ a $b = 2$. Anička v prvom ťahu skrúti špagát dĺžky 11 na dĺžku 3 (druhý špagát ostáva dlhý dva mačkometre). Bláčík nemá na výber, skrúti špagát dĺžky 3 o dva mačkometre, a Aničke zostanú špagáty s dĺžkami 1 a 2. Keď z dlhšieho odhryzne jeden mačkometer, Bláčíkovi ostanú dva špagáty dĺžky jedna. Nemá na výber, akýmkoľvek ťahom prehráva.

8. Algoritmické problémy

12 b za popis, 8 b za program

V nie až tak vzdialenej budúcnosti už všetkých prestalo baviť robiť pokusy na potkanoch, preto zvolili nové pokusné zviera: mačiatka. Mačiatka sú oveľa inteligentnejšie ako väčšina potkanov, preto aj výskum na nich môže byť oveľa komplikovanejší. Výskumníci na Matfyzě na katedre aplikovaných neľudských pokusov na zvieratách sa momentálne zaoberajú výskumom, ako mačiatka reagujú na riešenie rôznych algoritmických problémov.⁷

Vedci si všimli jednu pozoruhodnú vec. Mačiatka sú síce veľmi inteligentné, ale aj veľmi lenivé. Keď si zapnú contest, vyriešia ho takmer okamžite, ale nechce sa im kódovať.⁸ Preto skôr ako začnú kódovať, rozmyslia si mačky, aké známe algoritmy budú potrebovať na vyriešenie jednotlivých úloh. Všetky tieto algoritmy najprv naprogramujú, a potom ich už iba v každej úlohe použijú, teda nemusia nič kódovať viac krát.

Po dlhom a náročnom výskume vedci z katedry zistili pre každý známy algoritmus, koľko antipotešenia prinesie mačke, kým ho nakódi.⁹ Rovnako pre úlohy v conteste je ľahké povedať, koľko potešenia prinesie mačke, ak ju vyrieši a tiež aké algoritmy sú potrebné na jej vyriešenie. Všetok ostatný kód okrem algoritmov je z hľadiska potešenia irelevantný.

Keďže však algoritmické mačky sú drahé, chceli by vedci zistiť, koľko najviac potešenia by taká mačka mohla získať, ak by riešila nejaký contest. Keďže však túto úlohu nemôžu dať vyrábať mačkám, lebo – ako som povedal – sú drahé, uchýlili sa vedci k lacnejšej alternatíve: vám. Tak šup šup, makaj a počítaj!

Úloha

Dostanete popis jedného contestu a všetkých algoritmov, ktoré mačky poznajú. Pre každý algoritmus máte zadané, koľko antipotešenia mačke prinesie, ak ho nakódi. Tiež pre každý problém contestu viete, koľko potešenia mačke prinesie, ak ho vyrieši, a ktoré algoritmy sú potrebné na jeho vyriešenie. Na vyriešenie problému potrebuje mačka nakódiť všetky tieto algoritmy. Všetko ostatné je z hľadiska potešenia irelevantné

Každý algoritmus stačí nakódiť raz, a každý problém sa dá vyriešiť najviac raz (keď mačka aj odovzdá viac krát to isté už ju to znovu nepoteší).

Vašou úlohou je zistiť, koľko najviac potešenia vie mačka za daný contest získať, teda maximálnu hodnotu súčtov potešenia za problémy mínus súčtov antipotešenia za algoritmy.

Formát vstupu

V prvom riadku vstupu sú čísla n, m ($1 \leq n, m \leq 200$) postupne udávajúce počet problémov v conteste a počet známych algoritmov.

V druhom riadku je n medzerou oddelených čísel a_1, \dots, a_n , pričom číslo a_i ($0 \leq a_i \leq 10^9$) určuje počet získaného potešenia za vyriešenie problému i .

V treťom riadku je m medzerou oddelených čísel b_1, \dots, b_m , pričom číslo b_j ($0 \leq b_j \leq 10^9$) určuje počet antipotešenia za nakódenie j -teho algoritmu.

Nasledujúcich n riadkov popisuje potrebné algoritmy na vyriešenie každého z problémov,

⁷Sme v budúcnosti kde už aj mačky zvládnu hitnúť GM na codeforces.

⁸Toto správanie je nadmieru podobné matfyz tímom súčasnosti

⁹Aj tie najschopnejšie mačky občas zomierajú keď majú nakódiť intervaláč

i -ty z nich začína číslom k_i ($0 \leq k_i \leq m$) udávajúcím počet potrebných algoritmov pre i -ty príklad. Za ním nasleduje k_i medzerou oddelených čísel c_1, \dots, c_{k_i} ($0 \leq c_j \leq m$), j -te z nich znamená, že na vyriešenie problému i treba nakódiť algoritmus c_j .

Formát výstupu

Na jediný riadok vypíšete maximálne potešenie, ktoré môže mačka za zadaný contest dosiahnuť. Všimnite si, že výsledok bude nezáporný.

!POZOR! výsledok sa nemusí zmestiť do bežnej celočíselnej premennej, odporúčame použiť napríklad premenné typu `long long` v `c++`.

Hodnotenie

Sú 4 sady testovacích vstupov, každá za 2 body. Pre jednotlivé sady platia nasledujúce obmedzenia:

Sada	1	2	3	4
$1 \leq n, m \leq$	20	40	60	200

Príklady

vstup

```
3 4
9 8 9
5 3 4 10
2 1 2
2 2 3
2 3 4
```

výstup

```
5
```

Mačka naprogramuje problémy 1 a 2, čo jej dá 17 potešení. Na to ale musí naprogramovať algoritmy 1, 2 a 3, čo jej dá 12 antipotešení. Spolu teda získa 5 potešení.