



# Korešpondenčný seminár z programovania

Leták letnej 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 neprehrádzaj riešenia ostatným riešiteľom. Odpisovanie riešení a prezradenie riešení pred termínom kola je porušením pravidiel KSP. Po skončení kola sa, samozrejme, o riešeniach 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 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**. 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 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).

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 2. kola letnej časti

Termín odoslania riešení tohto kola je pondelok **3.8.2020**. Doprogramovávanie končí v podneľok 17.8.2020.

### 1. Polovica párna

12 b za popis, 8 b za program

Za posledné obdobie mal Kúl Spolok Pandemiológov naozaj kopu práce. Popri bežných povinnostiach zvládali rozbehnúť nový výskum zameraný na boj s vírusmi. Po niekoľkých mesiacoch tvrdej driny sa dostavili veľmi dobré výsledky. Vytvorili metódu, podľa ktorej dokážu na vírus jednoducho vytvoriť spoľahlivú vakcínu, a tak ho zneškodniť. Spôsob je veľmi jednoduchý. Každý vírus má svoje charakteristické číslo  $n$ . Na to, aby našli vakcínu k danému vírusu stačí nájsť  $n$  takých prirodzených čísel, že prvých  $n/2$  čísel je párnych a druhých  $n/2$  nepárnych. Zároveň musí platiť, že súčet prvej polovice je rovný súčtu druhej polovice. Kým ale celý KSP odišiel na konferencie prezentovať tento úžasný vynález, tebe nechali na starosť kopu vírusov, na ktoré máš nájsť vakcínu. . .

#### Úloha

KSP ti nechalo  $t$  vírusov, a ty máš pre každý z nich povedať či naň existuje vakcína. Ak existuje, tak takú nejakú vakcínu máš vypísať. Pre všetky čísla  $n_i$ , pre ktoré existuje vakcína máš vypísať presne  $n_i$  medzerou oddelených čísel takých, že súčet prvej polovice čísel sa rovná súčtu druhej polovice. Všetky čísla musia byť unikátne a všetky čísla v prvej polovici musia byť párne a v druhej nepárne. Takýchto možností je samozrejme veľa a môžete vypísať ľubovoľnú z nich.

#### Formát vstupu

Na prvom riadku vstupu dostanete číslo  $t$ , ( $t \leq 100$ ), ktoré označuje počet vírusov. Nasleduje  $t$  riadkov, na každom je práve jedno párne číslo  $n_i$ , ( $n_i \leq 10\,000$ ), ktoré označuje vírus.

#### Formát výstupu

Pre každé  $n_i$  na samostatný riadok vypíšte “ano” alebo “nie”, podľa toho či sa dá urobiť pole dĺžky  $n_i$  také, že spĺňa podmienky zo zadania. V prípade ak vypíšete “ano”, na nový riadok vypíšte aj toto pole. Teda, vypíšete  $n_i$  medzerou oddelených čísel  $a_{i,j}$ . Dodržte aby  $0 < a_{i,j} < 1\,000\,000$ , a každé číslo bolo unikátne (nachádzalo sa v Tebou vypísanej postupnosti čísel len raz).

#### Príklad

vstup

```
5
10
2
4
8
6
```

výstup

```
nie
nie
ano
2 4 1 5
ano
2 6 4 8 5 3 1 11
nie
```

*Samozrejme, toto nie sú jediné možnosti, ktoré mohli byť vypísané za “ano”. Napríklad za prvým “ano” rovnako mohlo byť aj “6 10 11 5”, alebo “4 2 5 1”.*

### 2. Atypické zasadnutie

12 b za popis, 8 b za program

Predseda Kúl Spolku Pandemiológov sa v rámci aktuálnej krízy rozhodol zvolať zasadnutie odborníkov zo všetkých krajov. Takéto zasadnutie je veľmi prestížne a prirodzene tak má určité absurdné pravidlá. Jedno z nich je napríklad to, že sa zasadá vždy výlučne za okrúhlim stolom a každý kraj má pri danom stole svoje pevne stanovené miesto. Preto každý, kto si sadne na miesto daného kraja, musí rozprávať iba o tom konkrétnom

kraji. Každý odborník, ktorý sedí na mieste svojho kraja je tak výrazne efektívnejší ako keď sedí na mieste iného kraja o ktorom nič nevie.

Odborníci sú tvrdohlavé stvorenia a za stôl si sadajú v poradí, v ktorom prišli do miestnosti a odmietajú si presadnúť. Predseda avšak potrebuje presné informácie z čo najviac krajov a tie mu vie dať iba odborník z daného kraja. Jedinou možnosťou ako tento stav dosiahnuť, je otočiť stôl o niekoľko miest do ktorejkoľvek strany tak, aby čo najviac odborníkov sedelo na svojom mieste. A keďže stôl je veľmi ťažký, ideálne ho chce otočiť o čo najmenej miest.

### Úloha

Tvojou úlohou je pomôcť predsedovi určiť o koľko najmenej miest treba otočiť okrúhly stôl tak, aby čo najviac odborníkov sedelo na svojom mieste. Predsedu taktiež zaujíma koľko odborníkov tak bude sedieť na svojom mieste. Odborníci si sadajú za stôl v poradí v akom prišli, odborník  $n_i$  si tak vždy sadne na pozíciu  $i$ . Odborník sedí na svojom mieste vtedy, ak je z kraja  $i$  a zároveň sedí na mieste  $i$  pri stole. Stôl je možné otočiť do oboch strán.

### Formát vstupu

Na prvom riadku vstupu dostanete číslo  $n$ , ( $1 \leq n \leq 10^6$ ), ktoré označuje počet krajov z ktorých prichádzajú jednotliví odborníci. Na druhom riadku sa nachádza  $n$  čísiel  $n_i$ , ( $1 \leq n_i \leq n$ ) označujúcich kraj z ktorého pochádza  $i$ -ty odborník. Každé číslo od 1 po  $n$  sa na vstupe nachádza práve raz.

### Formát výstupu

Na výstup vypíš dve čísla,  $k$  - posun stola,  $t$  - počet odborníkov na svojom mieste, tak aby čo najviac odborníkov sedelo na svojom mieste.

### Príklady

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

*Ak otočíme stôl o dve miesta, odborníci 3, 4 a 5 budú sedieť na svojom mieste.*

vstup	výstup
<pre>4 4 1 2 3</pre>	<pre>1 4</pre>

*Prvý prišiel odborník zo štvrtého kraja, následne už prišli odborníci po poradí. Stôl nám tak stačí otočiť o jedno miesto aby všetci odborníci sedeli na mieste so správnym číslom.*

vstup	výstup
<pre>4 2 3 4 1</pre>	<pre>1 4</pre>

*Podobne ako pri predchádzajúcom príklade nám stačí otočiť stôl iba o jedno miesto, tento krát avšak do druhej strany.*

## 3. Na dušu liek

12 b za popis, 8 b za program

Mesto je v karanténe. Všetci sú doma odizolovaní a odlúčení od akejkoľvek ľudskej interakcie. Obyvatelia už dva mesiace nevideli žiaden úsmev, lebo všetky tváre sú zahalené v rúškach. Počasie je ponuré, doma niet čo robiť. Všetci sú v depresii.

Mestská správa však nestráca nádej a rozhodla sa s tým niečo urobiť. Obyvateľstvo by určite rozveselilo trocha motivačnej hudby z rádia. V meste však nie je vysielateľ, tak treba nejaký postaviť.

Avšak so zvyšujúcou sa vzdialenosťou od vysielateľa sa zhoršuje signál, a zvyšuje depresia. Presnejšie depresia každého občana je rovná druhej mocnine jeho vzdialenosti od vysielateľa v metroch.

Aby sa signál mohol lepšie šíriť do všetkých strán a nestáli mu v ceste budovy rozhodli sa ho postaviť na križovatke. Ktorá križovatka však bude najvhodnejšia?

### Úloha

Mesto má tvar štvorcovej siete o rozmeroch  $r \times s$ . Políčka sú štvorcové pozemky s dĺžkou strany 8 metrov. Každý pozemok má súradnice  $(i, j)$ , je na ňom jeden dom a žije v ňom  $c_{ij}$  ľudí. Pre zjednodušenie predpokladáme,

že všetci ľudia sa nachádzajú presne v strede pozemku na ktorom bývajú. V severozápadnom rohu mesta je pozemok so súradnicami  $(1, 1)$  a v juhovýchodnom  $(r, s)$ . Medzi pozemkami vedú ulice, ich šírku môžeme zanedbať. Tieto ulice sa pretínajú v  $(r + 1) \times (s + 1)$  križovatkách. V severozápadnom rohu mesta je križovatka so súradnicami  $(0, 0)$ , v juhovýchodnom  $(r, s)$ .

Zistite na ktorej križovatke má mesto postaviť vysielač aby celková depresia jeho obyvateľstva bola najmenšia možná.

### Formát vstupu

Na prvom riadku sú dve celé čísla  $r$  a  $s$ , oddelené medzerou - rozmery mesta. Platí  $1 \leq r, s \leq 1000$ . Nasleduje  $r$  riadkov, každý z nich obsahuje  $s$  celých čísel oddelených medzerou. V  $i$ -tom riadku a  $j$ -tom stĺpci je číslo  $c_{ij}$ , počet obyvateľov domu na súradniciach  $(i, j)$ . Platí  $0 \leq c_{ij} \leq 100\,000$ .

Je osem testovacích sád. V sádach 1 a 2 navyše platí  $r, s \leq 30$ , a v sádach 3, 4 a 5 platí  $r, s \leq 200$ .

### Formát výstupu

Na prvý riadok vypíšte jedno číslo; najmenšiu dosiahnuteľnú depresiu. Na druhý riadok vypíšte súradnice vysielača, ktoré túto hodnotu dosiahnu. Ak existuje viac riešení vypíšte ľubovoľné z nich.

Výsledok sa nemusí zmestiť do klasickej 32 bitovej číselnej premennej (v C++ preto použite premennú typu `long long`).

### Príklady

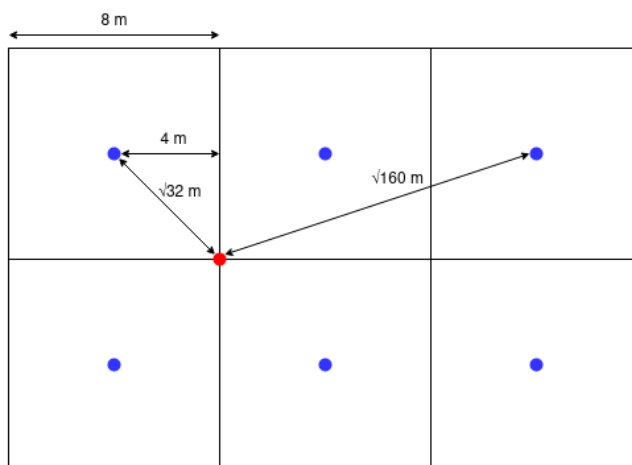
vstup

```
2 3
1 2 2
2 9 1
```

výstup

```
928
1 1
```

Optimálne je tiež aj  $(1, 2)$ .  $928$  sme dostali ako  $32 \cdot (1 + 2 + 2 + 9) + 160 \cdot (2 + 1)$



vstup

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

výstup

```
2880
2 2
```

## 4. Dezinfekcia

12 b za popis, 8 b za program

Krajské Mesto Pravičiarov (KMP) čelí architektonickej výzve. Z dlhodobých pozorovaní obyvatelia zistili, že počas roka sa vyskytujú, čím ďalej, tým výraznejšie extrémny, čo sa týka množstva pitnej vody v ich oblasti. Niekedy majú pitnej vody veľa a inokedy veľmi málo. Preto prišli s nápadom, že potrebujú zväčšiť mestskú nádrž na vodu, aby počas priaznivejšieho obdobia dokázali uskladniť prebytočnú vodu na horšie časy.

Nádrž sa nachádza v podzemí mesta tvoreného skalami<sup>1</sup>. Ich podzemie je tiež zaujímavé tým, že skaly je

<sup>1</sup>Napriek tomu je podložie mesta stabilné. Áno, je to divné, ale funguje to, tak sa do toho nestarajme.

možné odstraňovať (pri pohľade z hora) iba po štvorcových kusoch s rozmermi  $1m \cdot 1m$ . Nádrž je tvorená iba súvislým voľným priestorom medzi skalami. Skaly sú vodotesné a teda cez ne nepresakuje voda.

Jeden by si povedal, že tu o nič nejde. Stačí bezhlavo vykopať väčšiu dieru. No ako to už v takýchto KSP úlohách býva, je tu háčik - v podzemí sa nachádzajú aj viaceré menšie nádrže na dezinfekciu, ktoré boli vytvorené iba nedávno. Problém je teda v tom, že obyvatelia neobľubujú konzumáciu dezinfekčných prostriedkov. Preto je pri plánovaní rozšírenia nádrže potrebné dbať na to, aby sa dezinfekcia nezmiešala s pitnou vodou.

Analytici z KMP zistili, že je potrebné, aby bola plocha nádrže zväčšená aspoň na  $s$  metrov štvorcových. V úlohe projektovania nových stavebných úprav si však až tak neveria. Preto ste tu vy. Pomôžte obyvateľom KMP vyriešiť tento problém!

## Úloha

Poznáme pôdorys podzemia - 2D štvorcová mriežka, kde jeden štvorček má rozmery  $1m \cdot 1m$ . Odstráňte z neho najmenší možný počet skál tak, aby vo výslednom pôdoryse zaberala nádrž na pitnú vodu aspoň  $s$  metrov štvorcových a aby sa do nej nevyliat obsah niektorej nádrže na dezinfekciu.

Každá nádrž na dezinfekciu je reprezentovaná znakom D a má rozmery presne  $1m \cdot 1m$ , čiže okolo každého znaku D sa na každom z existujúcich štyroch susedných políčok nachádza skala (#).

Nádrž na pitnú vodu je tvorená políčkami označenými znakom . . Všetky políčka nádrže na vodu tvoria práve jednu súvislú nádrž. Inými slovami, z každého políčka nádrže na pitnú vodu je možné sa dostať na každé políčko nádrže na pitnú vodu prechodom iba cez políčka nádrže na pitnú vodu.

Výsledná nádrž musí byť takisto súvislá. Obyvatelia nebudú spokojní s riešením, ktoré zapríčiní existenciu viac ako jednej nádrže na pitnú vodu v podzemí ich KMP.

Vypíšte mapu podzemia, ktoré spĺňa zadané požiadavky!

## Formát vstupu

Na prvom riadku vstupu sa nachádzajú 3 čísla. Rozmery mapy podzemia - počet riadkov  $m$  a počet stĺpcov  $n$  - a požadovaná plocha nádrže  $s$ . Platí, že  $1 \leq m, n \leq 1000$  a  $1 \leq s \leq m \cdot n$ . Nasleduje  $m$  riadkov po  $n$  znakov reprezentujúcich mapu. Môžu sa tu vyskytovať iba znaky . (nádrž na pitnú vodu), # (skala) a D (dezinfekcia).

## Formát výstupu

Na výstup vypíšte mapu, ktorá vznikla splnením úlohy! Vypíšte ju v rovnakom formáte, v akom bola na vstupe -  $m$  riadkov po  $n$  znakov!

Pre jeden vstup môže existovať viacero správnych riešení. V takom prípade vypíšte ľubovoľné z nich!

Môžete počítať s tým, že pre každý testovací vstup existuje riešenie.

## Príklady

vstup

```
6 6 10
###.##
#D#..#
###..#
##.##
##.###
#####
```

výstup

```
##.##
#D#..#
##..#
##.##
##.###
#####
```

*Jedným riešením je odstránenie skál v rohoch okolo D. (Ak sa skaly dotýkajú hranami, spojenie je stále vodotesné a obyvatelia nebudú otrávení.)*

vstup

```
6 6 20
###.##
#D#..#
###..#
##.##
##.#D#
#####
```

výstup

```
##...
#D#...
.#...
...#.
...#D#
#####
```

## 5. Éra krátkych skratiek

12 b za popis, 8 b za program

Janko sa začal učiť pracovať s Unixovým terminálom. Ako prvý sa naučil príkaz `echo` s veľmi príznačným menom, ktorý ako ozvena vypíše svoje parametre na štandardný výstup.

Potom však došiel k príkazom, ktoré začali mať príliš krátke a nezrozumiteľné mená. Začínalo to úplne nebadane, príkazmi `cp`, `mv` a `rm` na kopírovanie, presúvanie a mazanie súborov, ktorých mená vznikli jednoducho z `CoPy`, `MoVe` a `ReMove`. Neskôr sa to začalo zhoršovať. Janko sa dozvedel napríklad o príkazoch `chmod` (change mode), `tar` (tape archiver), `tr` (transform) či `chgrp` (change group). Janko si príkazy nedokázal zapamätať a v skriptoch, ktoré napísal, sa už chvíľu nato nedokázal vyznať. Ale malo to aj svoje výhody: menej búchania do klávesnice, menej prenesených mobilných dát<sup>2</sup>, ..., a tu to aj skončilo.

Kto si má všetky tie skratky pamätať, pomyslel si Janko a začal sa radšej učiť programovať v jazyku `C`. Vôbec si však nepomohol, práve naopak. Tu mená funkcií boli nielen prikrátke, ale priam až kryptické. Napríklad `strchr`. Po dlhých minútach zamyslenia zistil, že to `str` znamená `string`, `r` znamená `reverse` a `chr` znamená `character` a funkcia teda hľadá výskyt znaku v reťazci odzadu.

Keďže ho však premýšľanie začalo bolieť, rozhodol sa už druhýkrát zmeniť predmet svojho záujmu. Vybral si jazyk `SNOBOL`. Ale toto už Jankovi nedalo. Prečo práve `SNOBOL`? Ako k tomu došli? Vraj `StrINg Oriented symBOLic Language`. To nijak nedáva zmysel.

Napokon došiel k tomu, že vo všeobecnosti skratka vzniká tak, že len jednoducho vyberieme niektoré znaky názvu, pričom zachováme ich poradie a zapíšeme ich za seba. Z každého slova musíme vybrať aspoň jedno písmeno. A žiadne ďalšie pravidlá zrejme neplatia.

Janko by chcel vedieť overovať, či je daná skratka platná. Ale rozmýšľanie ho už po lúštení `strchr` a `SNOBOLu` veľmi bolí, rád by to teda nechal na vás.

### Úloha

Vašou úlohou je spočítať počet možností, ako mohla vzniknúť skratka  $S$  z názvu  $T$ .

Aby skratka  $S$  mohla vzniknúť z textu  $T$ , musí platiť, že písmenám v  $S$  vieme priradiť písmená v  $T$ , pričom musíme zachovať poradie (teda ak priradíme písmenko z  $S$  nejakému písmenku z  $T$ , ďalšiemu môžeme priradiť len nejaké napravo v  $T$ ) a z každého slova  $T$  musíme použiť aspoň jedno písmeno.

Keďže počet možností je v niektorých prípadoch naozaj veľký, vypíšte len jeho zvyšok po delení  $10^9 + 7$ .

### Formát vstupu

Na prvom riadku sa nachádza číslo  $w$  – počet slov textu, z ktorého má byť utvorená skratka.

Na druhom riadku sa nachádza skratka - reťazec  $S$  a na treťom riadku sa nachádza text  $T$ , ktorého skratka to má byť. Oba reťazce na vstupe pozostávajú len z malých písmen anglickej abecedy a medzier. Jednotlivé slová  $T$  sú oddelené práve jednou medzerou (medzier je teda  $w - 1$ ). Skratka žiadne medzery neobsahuje.

### Formát výstupu

Vypíšte jedno číslo – počet možností, ako možno priradiť písmenám skratky  $S$  písmená textu  $T$  tak, aby tvorili platnú skratku, modulo  $10^9 + 7$ . Ak to nie je možné, vypíšte 0.

### Hodnotenie

Nech  $M$  je dĺžka skratky a  $N$  dĺžka textu. Sú 4 sady vstupov. V jednotlivých sádach platia nasledovné obmedzenia:

1.  $M \leq 5, N \leq 12$
2.  $M \leq 12, N \leq 50$
3.  $M \leq 120, N \leq 200$
4.  $M \leq 200, N \leq 10000$

### Príklady

vstup

```
2
tar
tape archiver
```

výstup

```
4
```

*Možnosti sú TApe aRchiver, TApe archiveR, TApe ARchiver a TApe ArchiveR.*

<sup>2</sup>Znižovanie množstva prenesených dát je naozaj dôvodom, prečo majú príkazy tak veľmi skrátene názvy. Kedysi, v časoch terminálov a telefónnych liniek, to naozaj malo význam.

vstup	výstup
4 snobol string oriented symbolic language	1
vstup	výstup
3 strchr string reverse char	3
vstup	výstup
4 radar radio detection and ranging	1
vstup	výstup
4 acm academy of computer makers	0

Jediná možnosť je *RA*dio *D*etection *A*nd *R*anging. *RA*Dio *d*etection *A*nd *R*anging nie je platná možnosť, pretože zo slova *detection* sme nepoužili žiadne písmeno.

Skratka je jednak príkrátka, a jednak neobsahuje žiadne z písmen slova *of*.

## 6. Mandarínske kráľovské problémy

12 b za popis, 8 b za program

V Mandarínsku žije samolúby kráľ Nenásytnan Veľký, ktorý má rád mandarínky. Má ich až tak rád, že každý deň počas mandarínkovej sezóny jednu zje<sup>3</sup>. A keďže on je veľmi nenásytný tak každý deň musí zjesť tú najväčšiu čo sa v predchádzajúci deň v Mandarínsku urodila. V Mandarínsku počas sezóny narastie každý deň  $n$  mandaríniek. Tieto mandarínky sa potom rovnomerne poukladajú na jednu dlhú linku, kde ich manuálne zoradujú pracovníci od najmenej po najväčšiu s cieľom aby sa najväčšia mandarínka dostala priamo na kráľovský stôl. Každý pracovník je zamestnaný na určitom úseku linky a jeho úlohou je usporiadať mandarínky na linke podľa veľkosti v rámci jeho úseku (rozsahu rúk).

Keďže aj do Mandarínska prišla pandémia, z hygienických dôvodov sa môže na každom pracovisku nachádzať naraz iba jeden zamestnanec. Podľa nariadenia hlavného hygienika Mandarínska sú zamestnanci každý deň púšťaní do práce postupne podľa ich zamestnaneckého čísla. Každý z nich tam môže byť iba nevyhnutný čas a potom musí ísť domov.

Tento postup musí dodržiavať aj mandarínková triediaca linka. Šéf tejto linky Kleofáš Múdry rozmýšľa ako ho najlepšie implementovať. Počas prvých dní pandémie do firmy púšťal všetkých zamestnancov postupne, tak ako to vyžaduje nariadenie. Potom si však uvedomil, že keďže jedinou úlohou celej linky je nájsť najväčšiu mandarínku a poslať ju ku kráľovi, možno by mohol ako dobrý šéf nechať niektorých svojich zamestnancov doma. Teraz rozmýšľa nad tým koľko zamestnancov reálne musí chodiť do práce aby linka plnila to na čo bola určená. Viete mu s tým pomôcť?

### Úloha

Na triediacej linke sa v pravidelných rozstupoch nachádza  $n$  mandaríniek. Vo firme pracuje  $m$  zamestnancov. Každý z nich príde do práce a podľa veľkosti vzostupne utriedi mandarínky na linke medzi  $a_i$ -tou a  $b_i$ -tou vrátane, pričom na linke udržiava pravidelné rozstupy. Kleofáš chce vybrať čo najmenej zamestnancov tak, aby vedel zaručiť že ak by prišli do práce iba oni, tak nech je najväčšia mandarínka na začiatku dňa na linke na ľubovoľnom mieste, na konci dňa vždy skončí na pozícii  $n$ . Zistíte koľko najmenej zamestnancov musí vybrať do smeny. Nezabudnite že do práce musia prísť v poradí podľa zamestnaneckého čísla.

### Formát vstupu

Na prvom riadku vstupu sa nachádzajú dve celé čísla  $n, m$  ( $1 \leq n, m \leq 300\,000$ ) – denná produkcia Mandarínska a počet zamestnancov na linke. Na ďalších  $m$  riadkoch sa nachádzajú intervaly na ktorých pracujú jednotliví zamestnanci. Na  $i$ -tom riadku sa nachádzajú čísla  $a_i$  a  $b_i$  ( $1 \leq a_i \leq b_i \leq n$ ), kde  $[a_i, b_i]$  je interval ktorý je triedený zamestnancom so zamestnaneckým číslom  $i$ .

<sup>3</sup>Voľakedy ich jedol viac, ale to mu jeho diétny asistent zakázal, pretože sa nezmestil na kráľovskú fotografiu.



## Formát výstupu

Na výstup vypíšete jediné číslo – minimálny počet zamestnancov, ktorý musia prísť do práce na to, aby sa najväčšia mandarínka zaručene dostala na poslednú pozíciu na triediacej linke. V prípade že je na triediacej linke už teraz taký zmätok, že aj bez vynechania zamestnanca sa najväčšia mandarínka nie nutne dostane ku kráľovi, vypíšete  $-1$ .

## Hodnotenie

Sú 4 sady vstupov. V jednotlivých sadách platia nasledovné obmedzenia:

1.  $n \leq 10, m \leq 10$
2.  $n \leq 100, m \leq 100$
3.  $n \leq 300\,000, m \leq 2\,000$
4.  $n \leq 300\,000, m \leq 300\,000$

## Príklady

vstup	výstup				
<table border="1"><tr><td>6 2</td></tr><tr><td>1 4</td></tr><tr><td>4 6</td></tr></table>	6 2	1 4	4 6	<table border="1"><tr><td>2</td></tr></table>	2
6 2					
1 4					
4 6					
2					

Obaja zamestnanci musia prísť do práce, inak by sa najväčšia mandarínka z pozície 3 nemohla dostať na koniec linky na pozíciu 6.

vstup	výstup				
<table border="1"><tr><td>6 2</td></tr><tr><td>4 6</td></tr><tr><td>1 4</td></tr></table>	6 2	4 6	1 4	<table border="1"><tr><td>-1</td></tr></table>	-1
6 2					
4 6					
1 4					
-1					

V tomto prípade ak sa najväčšia mandarínka nachádza na pozícii 3 tak keď príde do práce prvý zamestnanec tak ju neposunie a druhý zamestnanec ju posunie iba na pozíciu č. 4. Aj keď prídu do práce všetci zamestnanci, najväčšia mandarínka na konci dňa nemusí byť na poslednej pozícii.

vstup	výstup					
<table border="1"><tr><td>6 3</td></tr><tr><td>1 4</td></tr><tr><td>2 5</td></tr><tr><td>3 6</td></tr></table>	6 3	1 4	2 5	3 6	<table border="1"><tr><td>2</td></tr></table>	2
6 3						
1 4						
2 5						
3 6						
2						

Stačí ak do práce príde iba prvý a tretí zamestnanec. Ak najväčšia mandarínka začínala medzi pozíciami 1 až 4, prvý zamestnanec ju umiestni na pozíciu 4. Tretí zamestnanec ju stadiť vie položiť na koniec, a zvláda aj prípad kedy mandarínka bola na začiatku na pozícii 5.

## 7. Istota čistoty

12 b za popis, 8 b za program

V T2 je špinavo. Dávid tam síce z času na čas povysáva, no Marcelovi to nestačí. Ako správny zdravotník si je vedomý toho, že pre odstránenie mikróbov bude nutné použiť niečo silnejšie. Doniesol si teda svoj tepovač a všetky koberce za doobedie dôkladne vyčistil.

Vtom si ale spomenul na ďalšiu nepríjemnosť. Totiž, vo vlhkom prostredí sa dobre darí plesniam. Zobral teda koberce a dal ich sušiť na slnečný trávnik. Trávnik je však malý a koberce sa nezmestia vedľa seba. Niektoré teda naukladal jeden na druhý. Nechcel ale, aby to bolo až príliš chaotické, nuž sa žiadne dva koberce nedotýkajú stranami ani rohmi. Môže však byť menší koberec položený na väčšom, alebo naopak.

Na slnečnom trávniku sa zvyknú hrať deti. Nejaké koberce im vôbec neprekážajú. Keďže ale boli mokré, jedno z detí sa pošmyklo a rozsypalo po celom trávniku lentilky. Voda v kobercoch začala rozpúšťať farebnú cukrovú polevu. Pod každou lentilkou farba z polevy presiakla cez všetky koberce až po trávnik.

Keď si tú hrôzu Marcel všimol, prišlo mu zle. Potom si ale uvedomil, že sa vlastne nič zlé nestalo a začal namiesto toho rozmýšľať, koľko rôznych lentilkových farieb je na každom z kobercov. Je ale veľmi unavený z toľkého úmorného tepovania. Keby mu len niekto s touto úlohou pomohol...

## Úloha

Marcel vám povedal, ako rozmiestnil koberce. Koberce sa nepretínajú ani nedotýkajú stranami ani rohmi. Dva koberce môžu byť navzájom nad sebou iba ak je jeden z nich úplne obsiahnutý v druhom a nedotýkajú sa

stranami. Teda môže sa stať že je menší pod väčším, ale aj to, že je väčší pod menším a dokonca môže byť celá kopa kobercov medzi nimi (pričom každá dvojica v tejto kope spĺňa tieto pravidlá).

Koberce sú obdĺžnikové a sú poukladané tak, aby ich strany boli rovnobežné s osami  $x$  a  $y$ .

Ďalej vám povedal súradnice a farby jednotlivých lentiliek. Farieb môže byť mnoho, no môže byť viac lentiliek tej istej farby.

Vašou úlohou je povedať Marcelovi, koľko rôznych farieb z lentiliek sa dostalo na ktorý koberec.

### Formát vstupu

Na prvom riadku vstupu sú dve čísla  $n$  a  $m$  - počet kobercov a počet lentiliek. Platí, že  $1 \leq n, m \leq 10^5$ .

Nasleduje  $n$  riadkov. Na  $i$ -tom z nich je popis  $i$ -teho koberca. Tento popis pozostáva zo 4 čísiel  $1 \leq x_1, y_1, x_2, y_2 \leq 10^9$  - súradníc ľavého dolného a pravého horného rohu. Môžete predpokladať, že  $x_1 < x_2$  a tiež  $y_1 < y_2$ .

Na  $j$ -tom z ďalších  $m$  riadkov je popis  $j$ -tej lentilky. Pozostáva z 3 čísiel  $1 \leq x, y, f \leq 10^9$  - súradníc lentilky a jej farby.

### Formát výstupu

Vypíšte  $n$  riadkov. Na  $i$ -tom z nich bude jedno číslo - počet rôznych farieb, ktoré lentilky pridali na  $i$ -ty koberec.

### Hodnotenie

Sú 4 sady vstupov. Platia v nich nasledovné obmedzenia:

Sada	1	2	3	4
$1 \leq n, m \leq$	1000	50000	100 000	100 000

### Príklad

vstup

```
3 3
1 1 10 10
2 2 9 9
4 4 6 8
2 10 1
2 9 2
5 7 3
```

výstup

```
3
2
1
```

## 8. Aminokyseliny

12 b za popis, 8 b za program

Šandyna je šikovná bioinformatička a vo svojej práci často manipuluje RNA a DNA rôznych vírusov. Naposledy potrebovala izolovať sekvenciu istého Krutého Sveta-Paralyzujúceho vírusu. Žiaľ, pri pokuse sa jej to nepodarilo, miesto toho len izolovala  $n$  úsekov aminokyselín a nie presnú sekvenciu ktorú by potrebovala. Ale ešte nič nie je stratené! V laboratóriu biologickými čarami vie Šandyna spájať sekvencie, aby dostala RNA vírusu. Ale to nie je len tak, spájať genetické sekvencie. Niektoré sa pripájajú ťažšie než iné. Potrebuje preto zistiť ako ich čo najľahšie spojiť do chceného RNA. Pomôžte jej!

### Úloha

Šandyna má reťazec  $T$ , RNA vírusu ktorý chce zostaviť. Tiež už má  $n$  rôznych úsekov genetického kódu. Genetický kód si vieme zapísať ako reťazec malých písmen anglickej abecedy. Šandyna má tiež vzorky nukleidov reprezentovaných každým písmenom. Šandyna začína s prázdnu sekvenciou a tú výslednú by chcela zostrojiť pomocou týchto štyroch postupov:

- Na začiatok aktuálnej sekvencie prilepí jeden nukleid (písmenko). Prilepenie nukleidu  $c$  má cenu  $cl_c \cdot |S|$ .
- Na koniec aktuálnej sekvencie prilepí jeden nukleid (písmenko). Prilepenie nukleidu  $c$  má cenu  $cr_c \cdot |S|$ .
- Na začiatok aktuálnej sekvencie prilepí genetickú sekvenciu číslo  $i$  ( $1 \leq i \leq n$ ). Prilepenie sekvencie má cenu  $kl_i \cdot |S|$ .

- Na koniec aktuálnej sekvencie prilepí genetickú sekvenciu číslo  $i$  ( $1 \leq i \leq n$ ). Prilepenie sekvencie má cenu  $kr_i \cdot |S|$

Kde  $|S|$  je dĺžka aktuálnej sekvencie. Všimnime si, že pridať prvý nukleid/sekvenciu je vždy zadarmo.

Šandyna nevie vytvárať paralelne viac sekvencií a potom ich prilepiť k sebe. Ako najlacnejšie môže vytvoriť správny genetický kód?

### Formát vstupu

Na prvom riadku je číslo  $n$  - počet predom izolovaných sekvencií.

Na ďalších  $n$  riadkoch sú tieto sekvencie,  $p_1$  až  $p_n$ , reprezentované ako reťazce malej anglickej abecedy.

Na ďalšom riadku je 26 medzerami oddelených cien pripojenia jediného nukleidu na začiatok:  $cl_a$ , až  $cl_z$ .

Na ďalšom riadku je 26 medzerami oddelených cien pripojenia jediného nukleidu na koniec:  $cr_a$ , až  $cr_z$ .

Na ďalšom riadku je  $n$  medzerami oddelených cien pripojenia genetických sekvencií na začiatok:  $kl_1$ , až  $kl_n$

Na ďalšom riadku je  $n$  medzerami oddelených cien pripojenia genetických sekvencií na koniec:  $kr_1$ , až  $kr_n$

A na poslednom riadku je reťazec  $T$  - genetická sekvencia ktorú Šandyna chce vytvoriť.

### Formát výstupu

Vypíšte jediný riadok: najmenšiu cenu za ktorú vie Šandyna sekvenciu vytvoriť.

### Hodnotenie

Vo všetkých vstupoch platia nasledovné limity:

- $1 \leq |T| \leq 1000$
- $0 \leq n \leq 10000$
- Všetky počiatočne izolované genetické sekvencie  $p_i$  majú dĺžku najviac 100.
- Všetky ceny sú medzi 1 a  $10^9$
- Všetky reťazce obsahujú iba malé písmená anglickej abecedy

Sú 4 testovacie sady. V nich navyše platia nasledovné obmedzenia:

Sada	1	2	3	4
$n \leq$	0	10	10 000	100 000

V sade 2 navyše platí  $|p_i| \leq 10$ .

### Príklady

**POZOR!** Vstup v príkladoch je zalomený, len aby vošiel na stranu. Vstup pre váš program, bude v zadanom formáte.

vstup

```
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26
10 1 1 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26

abaca
```

výstup

```
15
```

Najlepšie je budovať sekvenciu odzadu. Cena je tak  $0 + 3 \cdot 1 + 1 \cdot 2 + 2 \cdot 3 + 1 \cdot 4 = 15$

vstup

```
3
aba
ba
xy
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 16 17 18 19 20 21 22 23 24 25 26
3 2 1 3 5 9 10 11 11 9 9 8 7 6 5 1
 33 22 11 90 1 1 2 3 5 8
1 2 3
1 1 1
abacaba
```

výstup

5

*Šandyna by mala začať s nukleoidom c. Potom za cenu  $1 \cdot 1 = 1$  by mala pripojiť sekvenciu 'aba' na začiatok a napokon tú istú sekvenciu aj na koniec (za cenu  $1 \cdot 4 = 4$ , keďže sekvencia po druhom kroku bola 'abac'). Celková cena je teda 5.*