Korešpondenčný seminár z programovania
XXXIV. ročník, 2016/17
Katedra základov a vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

Úlohy 1. kola letnej časti

Termín odoslania riešení tohto kola je pondelok **13. marca 2017**.

1. Žrúti

kat. Z; 6 b za popis, 4 b za program

V prísne tajnej KSP miestnosti T2 je koberec. Kedysi vraj býval modrý, teraz však už pohltil toľko prachu, omrviniek, vlasov a kadečoho ďalšieho, že je skôr sivý. Kubík sa teda rozhodol, že je najvyšší čas koberec opäť povysávať. S hrôzou však zistil, že za tú dobu, čo tam koberec ležal, na ňom začalo všeličo rásť. Dokonca sa v ňom vytvoril samostatný ekosystém! Spozoroval dva typy organizmov: predátorov a korisť. Predátori sú (prekvapivo) špecifickí tým, že žerú korisť. Korisť je špecifická tým, že sa množí.

Kubík má teraz dilemu; nemôže predsa vysávať ekosystém! Radšej by bol, keby ekosystém prirodzene vymrel. Totiž, keby sa stalo, že predátori požerú všetku korisť, nebudú mať čo jesť a vymrú. Ak sa to nestane, bude musieť zavolať deratizátorov. Je ochotný chvíľu s vysávaním počkať, ale má to vôbec zmysel? Čo ak sa ekosystém rozrastie a potom sa ho nezbaví už nikto?

Úloha

Pre jednoduchosť budeme náš ekosystém modelovať po kolách. V každom kole sa postupne udeje nasledovné:

- Každý predátor zožerie p kusov koristi. V prípade, že na toto nežije dostatočne veľa koristi, predátori vyžerú všetku žijúcu korisť a v ďalšom kole pomrú od hladu.
- Každá korisť, ktorá ostala, sa nahradí k novými korisťami.

V ekosystéme môžu nastať tri prípady:

- Po nejakom (konečnom) počte kôl vymrie všetka korisť a následne aj predátori. Vtedy hovoríme, že ekosystém *vymrie*.
- Ekosystém bude schopný fungovať donekonečna a množstvo koristi bude rásť nad všetky medze. Vtedy hovoríme, že ekosystém *rastie*.
- Ekosystém síce bude žiť donekonečna, ale existuje nejaké číslo H také, že množstvo koristi nikdy neprekročí H. Vtedy hovoríme, že ekosystém je v *rovnováhe*.

Zistite, ktorý z týchto prípadov nastane.

Formát vstupu

V prvom riadku sú štyri celé čísla a, b, p, k - začiatočný počet predátorov, začiatočný počet kusov koristi, počet koristi, ktorú za jedno kolo jeden predátor skonzumuje, a počet koristi, ktorá vznikne z jednej pôvodnej koristi (pôvodná korisť zanikne a bude nahradená k novými). Platí $1 \leq a, b, p, k \leq 10^4$.

Dávajte si pozor na veľkosť čísel, s ktorými pracujete. Keď napríklad vynásobíte tri čísla zo vstupu, výsledok sa vám nemusí zmestiť do 32-bitovej premennej.

Formát výstupu

Na výstupe vypíšte jedno slovo (bez úvodzoviek): “vymrie” , “rastie” alebo “rovnovaha” podľa toho, ktorý prípad nastane.

Príklady

vstup	výstup
`3 15 4 2`	`vymrie`

V ekosystéme sú traja predátori a 15 *kusov koristi. V prvom kole predátori zožerú* 12 *kusov koristi a zvyšné* 3 *sa rozmnožia na* 6*. V druhom kole už nie je pre predátorov dosť koristi, preto zožerú všetku a následne umrú.*

vstup

```
2 4 2 10
```

výstup

```
vymrie
```

Všetka korisť bude zožraná už v prvom kole.

vstup

```
2 5 2 10
```

výstup

```
rastie
```

Po prvom kole bude žiť 10 *kusov koristi, po druhom* 60*, ...*

vstup

```
1 2 1 2
```

výstup

```
rovnovaha
```

V každom kole najprv predátor zožerie jednu z dvoch koristí, následne sa tá druhá korisť rozmnoží opäť na dve.

2. Zajova paranoja

kat. Z; 6 b za popis, 4 b za program

Zajo sa hral príliš veľa počítačových hier a teraz sa mu o nich aj sníva. Najhoršie sú nočné mory, v ktorých sa ho snaží odstreliť ostreľovač. Keď sa totiž Zajo z takejto nočnej mory prebudí, nevie sa zbaviť pocitu, že naňho cez okno niekto mieri a po celý zvyšok noci už nezaspí.

Preto si chce dať posteľ do takej časti izby, v ktorej ho nemajú šancu trafiť. Samozrejme, bol by rád, ak by táto bezpečná časť izby bola čo najväčšia (ani s pocitom, že niekto mieri na miesto desať centimetrov od vašej postele sa nespí veľmi dobre).

Úloha

Zajova izba má tvar obdĺžnika, ktorého strany sú rovnobežné so základnými svetovými stranami a má okná smerom na sever a na východ. Tieto okná sú veľmi úzke (človek, ktorý staval Zajovu izbu bol zrejme tiež paranoický) a pre účely tejto úlohy ich budeme považovať za body na stranách nášho obdĺžnika.

Ostreľovači, o ktorých sa Zajovi sníva, naňho mieria vždy buď z kopca ležiaceho presne na sever od Zajovho domu, alebo z vysokej budovy na východ od Zajovho domu. Takže ak ho chcú odstreliť, musia strieľať rovnobežne so stenami izby.

Poznáte rozmery miestnosti a pozície jednotlivých okien. Nájdite súvislú časť Zajovej izby neohrozenú ostreľovačmi, s najväčším obsahom.

Formát vstupu

Na prvom riadku vstupu sú 4 celé čísla x, y, m, n, kde x a y sú rozmery izby (x je dĺžka severnej steny a y je dĺžka východnej steny), m je počet okien na severnej stene a n je počet okien na východnej stene. Platí $1 \leq x, y, m, n \leq 10^6$.

Na druhom riadku vstupu je m celých čísel $x_1, \ldots, x_m$: pozície jednotlivých okien na severnej stene. i-te okno je vo vzdialenosti x_i od severozápadného rohu Zajovej izby. Platí $0 < x_1 < x_2 < \cdots < x_m < x$ – okná sú teda zadané v poradí od západu na východ.

Na treťom riadku je n celých čísel $y_1, \ldots, y_n$: pozície jednotlivých okien na východnej stene. i-te okno je vo vzdialenosti y_i od juhovýchodného rohu izby. Platí $0 < y_1 < y_2 < \cdots < y_n < y$ – okná sú zadané v poradí od juhu na sever.

V 1. sade testovacích vstupov platí, že $m, n \leq 10$, v 2. sade $m, n \leq 1\,000$).

Formát výstupu

Vypíšte jeden riadok a v ňom hodnotu S, kde S je plocha (obsah) najväčšieho súvislého územia, na ktoré nemôže mieriť ostreľovač.

Môžete predpokladať, že S sa zmestí do bežnej **64-bitovej** premennej. V C++ použite `long long`, v Pascale `Int64`.

Príklady

vstup

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

výstup

```
10
```

vstup

```
5 5 1 1
1
1
```

výstup

```
16
```

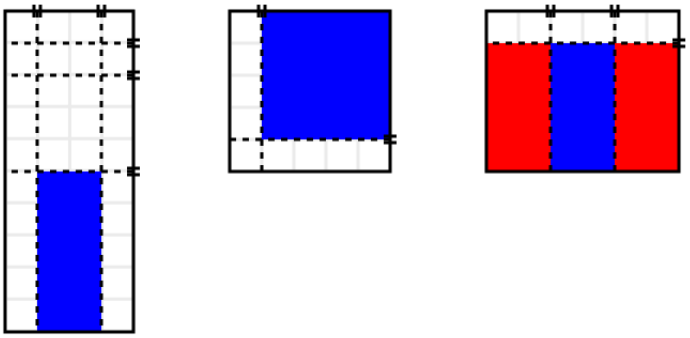
vstup

```
6 5 2 1
2 4
4
```

výstup

```
8
```

Situácie v jednotlivých príkladoch vyzerajú nasledovne (v poslednom príklade sú až tri rôzne časti s obsahom 8):



3. Zanedbané zastávky

kat. Z; 6 b za popis, 4 b za program

Kocúrkovo je to pravé mesto pre ľudí, ktorí sa vedia stratiť aj vo vlastnom byte. Je to totiž len jedna veľmi dlhá ulica. A keďže je taká dlhá, už roky tu jazdí mestská hromadná doprava.

Miestne autobusy jazdia po Kocúrkove tak, ako by ste od dlhej, rovnej ulice čakali. Pozdĺž ulice je postavených n autobusových zastávok. Každá autobusová linka má dve konečné zastávky, medzi ktorými jej autobusy jazdia hore-dolu, pričom stoja aj na každej zastávke po ceste. Kocúrkovo je však Kocúrkovo a tak sa môže stať, že niektoré linky začínajú aj končia na rovnakej zastávke (a teda ich autobusy iba stoja na tejto zastávke).

Kocúrkovské zastávky začali pod náporom neúprosného času hrdzavieť a chátrať. Preto si starosta Kocúrkova povedal, že ich dá všetky zrekonštruovať. A keď už sa prerábajú, tak by to malo byť vidno. Nech na to ľudia do volieb nezabudnú.

Rekonštrukcia jednej zastávky trvá jeden deň. Je možné rekonštruovať aj viac zastávok naraz. Starosta by teda mohol všetky zastávky nechať zrekonštruovať v ten istý deň, ale pre jeho volebné preferencie to nemusí byť tá najlepšia možnosť. Ľudia si totiž rekonštrukciu budú pamätať lepšie, ak bude prebiehať dlhšie.

Starostovi hrá do karát to, že Kocúrkovo je známe svojimi klebetníkni. Stačí, aby sa na trase autobusovej linky prerábala jediná zastávka a v autobuse vrava neutíchne až do večera. Ak sa teda v nejaký deň na trase každej linky prerába aspoň jedna zastávka, v celej Kocúrkovskej hromadnej doprave sa rozpráva iba o rekonštrukcii.

Úloha

Máte zadaný počet zastávok v Kocúrkove, počet liniek MHD a konečné zastávky, medzi ktorými tieto linky premávajú.

Starosta chce naplánovať rekonštrukciu zastávok tak, aby sa čo najdlhšie v autobusoch hovorilo iba o nej. Chce teda, aby od začiatku rekonštrukcie čo najdlhšie platilo, že na trase každej linky sa prerába aspoň jedna zastávka. Povedané ešte inak, starosta chce, aby sa čo najneskôr stalo, že na trase niektorej linky sa nebude prerábať žiadna zastávka.

A tu prichádzate na rad vy. Nájdite pre starostu Kocúrkova takýto plán a povedzte mu, koľko dní bude pri každej linke MHD aspoň jedna zastávka v rekonštrukcii. Pre každú zastávku teda naplánujte, v ktorý deň sa má prerábať.

Formát vstupu

V prvom riadku vstupu sú dve medzerou oddelené celé čísla n a m - počet zastávok a počet autobusových liniek. Platí $1 \leq n \leq 1\,000\,000$ a $1 \leq m \leq 100\,000$.

Nasleduje m riadkov, v každom z nich sú dve medzerou oddelené čísla z_i a k_i - poradové čísla konečných zastávok i-tej linky, od začiatku mesta. Platí $1 \leq z_i \leq k_i \leq n$.

Formát výstupu

Na prvý riadok výstupu vypíšte jedno číslo - po koľko dní od začiatku vieme rekonštruovať aspoň jednu zastávku na trase každej linky.

Na druhý riadok vypíšte n medzerou oddelených čísel - čísla dní, kedy sa majú rekonštruovať jednotlivé zastávky (v poradí od začiatku mesta po koniec). Rekonštrukcia sa začína nultým (0) dňom. Ak existuje viac optimálnych riešení, vypíšte ľubovoľné z nich.

Príklady

vstup

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

výstup

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

Na trase prvej linky sa zastávky rekonštruujú v deň 1, 0 *a* 2, *takže ľudia si budú rekonštrukciu všímať* 3 *dni. Podobne druhá linka. Pri tretej sa rekonštruuje v dni* 0 *a* 1. *Všetci si teda budú starostovu rekonštrukciu všímať aspoň* 2 *dni.*

4. Zakázané ohňostroje

kat. Z a O; 9 b za popis, 6 b za program

Konečne deň, na ktorý sa Emo celý rok tešil – *Silvester*. Tento rok ho chce stráviť na internáte v Bratislave. Keď sa to dozvedel jeho kamarát, neodpustil si štipľavú otázku, či už má nakúpené ohňostroje[1]. Emo iróniu nepochopil a tak, keďže nemal, začal narýchlo nejaké zháňať.

Zobral auto a šup ho na cesty. Ešte sa vo svojom rodnom meste[2] zastavil v obchode s pyrotechnikou a minul tam takmer celú výplatu. Odtiaľ už išiel priamo do Bratislavy.

Keď zišiel z diaľnice, zastavil sa ešte na benzínke. Ako tak tankoval, započúval sa do rádia a takmer dostal infarkt. Práve totiž išli správy o situácii v Bratislave:

- Za púšťanie rakiet v Bratislave hrozí veľká *pokuta*.
- Keďže policajti očakávajú v tento deň veľký počet výtržností[3], vymysleli si systém ako mať všetko pod kontrolou. Po meste na náhodné pevne určené miesta umiestnili **policajné kontroly**, kde kontrolujú obsah každého auta. Nemôžu dať síce pokutu za to, že niekto ohňostroj prepravuje, ale môžu si ho poznačiť a navštíviť večer.
- Aby bolo ťažké sa kontrole vyhnúť, na každej križovatke, kde nie je kontrola, stojí policajt ukazujúci len jeden prípustný smer jazdy. Každému, kto by z takejto križovatky chcel odísť iným, než povoleným smerom, dá pokutu (ale nechá ho ísť).

Emo si nejakú tú raketku neodpustí. Cesta na intrák tak preňho nebude iba o šoférovaní. Potrebuje v prvom rade zistiť **kade má na intrák ísť tak, aby minul čo najmenej peňazí na pokuty**.

Rýchlo preto zavolal Zajovi, nech mu pomôže. Zajo je totiž rodený Bratislavčan a má prístup k informáciám, o ktorých iní môžu len snívať. Potrebuje však vašu pomoc, aby vedel Emovi rýchlo odpovedať skôr ako si niekto všimne, že ešte aj telefonuje za volantom!

Úloha

Pre účely tejto úlohy budeme Bratislavu považovať za obdĺžnik $r \times s$ križovatiek. Na každej križovatke sa dá prejsť rovno, odbočiť vpravo, vľavo, aj otočiť sa.

Existujú križovatky nasledovných typov:

- *Križovatky s prikázaným smerom odjazdu*: tieto križovatky majú len jeden povolený smer odjazdu. Ak z nich Emo odíde iným ako povoleným smerom, dostane pokutu. Veľkosť tejto pokuty môže byť rôzna pre rôzne smery. Napríklad pre križovatky s prikázaným smerom odjazdu hore môže byť iná pokuta ako pre križovatky s prikázaným smerom odjazdu dole.

[1] v Bratislave sú totiž po novom zakázané.
[2] ktoré už trikrát vypálili, má 2 semafory (ak nerátame jeden nový pri hasičskej stanici) a 4 kruhové objazdy
[3] púšťania ohňostrojov

- *Križovatky s kontrolou*: tieto križovatky nemajú žiadne obmedzenia na to, ako cez ne môžete jazdiť, ale policajti na nich kontrolujú obsah kufra. Ak Emo počas cesty na internát prejde čo i len cez jednu takúto križovatku, o polnoci bude musieť zaplatiť pokutu za odpálenie rakety. Ak Emo prejde cez viacero takýchto križovatiek, bude platiť rovnako veľa, ako keby prešiel len cez jednu.
- *Križovatka s internátom*: táto križovatka je Emov cieľ. Prikázaný smer odjazdu z tejto križovatky nás nezaujíma, lebo Emo z tejto križovatky už nechce ísť ďalej.
- *Križovatka s benzínkou*: na tejto križovatke Emo začína a nie je tam ani kontrola, ani prikázaný smer odjazdu.
- Na niektorých políčkach obdĺžnika nie sú križovatky, ale *Dunaj*. Sem sa Emo s autom nikdy nechce dostať!

Dostanete popis situácie v Bratislave a aj veľkosti jednotlivých pokút. Zistite, koľko najmenej môže Emo zaplatiť na pokutách, ak chce prísť na internát a o polnoci tam odpáliť raketu. Počas svojej cesty nemôže opustiť Bratislavu.

Nepredpokladajte, že sa v Bratislave dá dostať na každú križovatku bez porušenia predpisov! Dokonca je možné, že na niektorých križovatkách je prikázaný smer jazdy do Dunaja.

Formát vstupu

Na prvom riadku vstupu dostanete medzerami oddelené postupne veľkosti pokút P_O, P_L, P_P, P_H, P_D: veľkosť pokuty za ohňostroj (ktorú Emo zaplatí, ak prejde cez nejakú kontrolu) a za porušenie prikázaného smeru odjazdu vľavo, vpravo, hore a dole. Všetky pokuty sú celé čísla z rozsahu 0 až 10^{12} (vrátane).

Nasleduje riadok s rozmermi mapy r, s ($1 \leq r \cdot s \leq 100\,000$). Ďalej nasleduje r riadkov po s znakoch popisujúcich samotnú Bratislavu. Každý z týchto znakov určuje typ jednej križovatky.

Znaky popisujúce križovatky môžu byť nasledovné: `K` – križovatka s kontrolou; `E` – križovatka s benzínkou, kde Emo začína; `I` – križovatka s internátom (cieľ); `L, P, H, D` – križovatky s prikázaným smerom odjazdu vľavo, vpravo, hore alebo na dole; `~` – Dunaj.

Môžete predpokladať, že na mape sa aj intrák aj Emo nachádzajú práve raz.

Formát výstupu

Vypíšte, koľko najmenej môže Emo zaplatiť na pokutách, ak sa chce dostať na intrák a o polnoci tam odpáliť raketu. V prípade, že neexistuje cesta vypíšte `-1`.

Hodnotenie a obmedzenia

Pre jednotlivé sady testov navyše platia nasledovné obmedzenia. Za vyriešenie každej sady získate 1 bod.

číslo sady	križovatky typu `K`	veľkosti pokút	rozmery mapy
1	na mape **nie** sú	pokuty sú 0	$2 \leq r \cdot s \leq 50$
2	na mape **nie** sú	pokuty sú buď 0 alebo 1	$2 \leq r \cdot s \leq 100\,000$
3	na mape **nie** sú	pokuty sú ľubovoľné	$2 \leq r \cdot s \leq 70$
4	na mape **nie** sú	pokuty sú ľubovoľné	$2 \leq r \cdot s \leq 100\,000$
5	na mape sú	pokuty sú ľubovoľné	$2 \leq r \cdot s \leq 10\,000$
6	na mape sú	pokuty sú ľubovoľné	$2 \leq r \cdot s \leq 100\,000$

Príklad

vstup

```
10 10 1 6 4
4 3
HLL
IPK
D~H
DLE
```

výstup

```
8
```

Optimálna cesta na internát vyzerá nasledovne: Z benzínky pôjde Emo doľava, pričom neplatí žiadnu pokutu, lebo na benzínke policajti nie sú. Potom pôjde opäť doľava, pričom opäť neplatí pokutu, lebo odišiel povoleným smerom. Následne pôjde dvakrát nahor, pričom oba razy poruší prikázaný smer jazdy nadol, teda dvakrát zaplatí pokutu 4. Keďže Emo neprešiel cez žiadnu kontrolu, o polnoci nedostane žiadnu pokutu.

vstup

```
1 2 3 4 5
2 2
E~
~I
```

výstup

```
-1
```

Emo môže rovno odparkovať, nikam sa nedostane.

vstup

```
1000 1 2 3 4
4 6
DHHHPK
LELKHK
~P~PKL
PKLHI~
```

výstup

```
1001
```

Kontrole sa cestou na internát nedá vyhnúť. Optimálne je ísť najprv trikrát doprava, potom hore, doprava, dvakrát dole, doľava a dole. Všimnite si, že po ceste Emo prejde až cez štyri kontroly, pokutu za odpaľovanie ohňostrojov však aj tak zaplatí iba raz. Okrem toho ešte poruší jeden prikázaný smer odjazdu vľavo, takže dokopy na pokutách zaplatí $1000 + 1 = 1001$.

5. Obezlička[4]

kat. Z a O; 8 b za popis, 7 b za program

Romanko, ako správny maturant, využil Vianočné prázdniny naplno. Posledný týždeň školy mu síce všetci učitelia pripomínali, ako dôležité je sa cez prázdniny venovať štúdiu, ale však čo – na to bude dosť času neskôr – pred Vianocami sa predsa ide nakupovať! No a po Vianociach si predsa tiež Romanko nebude kaziť náladu školou. Oveľa príjemnejšie je cez deň spať a v noci hrať počítačové hry. A po Silvestri? To je predsa perfektný týždeň na lyžovačku!

Na konci prázdnin si teda Romanko po veľmi "produktívne" strávenom čase pozrel, aké domáce úlohy to vlastne cez ne mal stihnúť. Otvoril si svoj zápisníček a so zhrozením zistil, že je tam práce skoro na tri týždne (aká to náhoda, však že?). Najťažšou úlohou je veľký projekt z informatiky. Čo teraz bude robiť? Ak svoj projekt z informatiky nespraví, bude mať len dve možnosti: odísť zo školy a zamestnať sa ako vývojár webových stránok, alebo dajako uprosiť svoju učiteľku informatiky, aby ho nechala prejsť s odretými ušami. V momentálnom stave však, napriek Romankovým vynikajúcim presvedčovacím schopnostiam, jeho projekt určite stačiť nebude.

Romanko si však spásonosne spomenul, že kedysi, keď namiesto vypracovávania dôležitých esejí riešil programátorské úlohy na SPOJi[5], mu pán profesor Demáček za každú vyriešenú úlohu napísal na vyvesenú tabuľu malé, bezvýznamné plus. Tieto pluská ale občas neboli vôbec malé (čím ťažšiu úlohu Romanko vyriešil, tým väčšie plus mu profesor napísal) a nemuseli by byť ani bezvýznamné – mohol by predsa svoju učiteľku presvedčiť, aby mu ich uznala ako bonusové body z programovania!

Utekal teda do Školského Výpočtového Laboratória a uprel pohľad na tabuľu s pluskami. Och, nie! Nejaký darebák ju nenávratne počmáral. Čo bude teraz robiť? Počmárané pluská mu učiteľka v žiadnom prípade neuzná! So slzami na krajíčku a niektorými cezeň si skleslo sadol na zem a prehodnocoval svoje životné rozhodnutia. Našťastie ste práve okolo Romanka išli vy. Vypočuli ste si jeho príbeh, a dali ste mu návrh – ak vám dá zvyšné horalky, ktoré ušetril pri podplácaní trpaslíkov[6], tak mu pomôžete spočítať, koľko plusiek ostalo na tabuli nepočmáraných. Romanko samozrejme nemal príliš na výber, a pristúpil na vaše podmienky.

Úloha

Tabuľa v Školskom Výpočtovom Laboratóriu je štvorcová a má stranu dĺžky n milimetrov. Pre potreby úlohy si ju vieme predstaviť ako mriežku pozostávajúcu z n riadkov. V každom riadku je n štvorčekových políčok o veľkosti 1×1 milimeter a každé z týchto políčok môže byť popísané alebo nepopísané.

Nepočmárané plus veľkosti v je štvorec so stranou dĺžky $2v + 1$ milimetrov, v ktorom je celý stredný riadok a stredný stĺpec popísaný, a všetky ostatné políčka v ňom sú nepopísané.

Za nepočmárané plus veľkosti v dostane Roman v bonusových bodov.

Každé nepočmárané plus veľkosti aspoň 2 obsahuje vo svojom strede menšie nepočmárané pluská. Za tieto menšie pluská však Roman už ďalšie body pochopiteľne nedostane.

Roman vám ukázal počmáranú tabuľu. Pomôžte mu zistiť, koľko bonusových bodov vie získať!

[4] http://slovnik.azet.sk/pravopis/slovnik-sj/?q=obezli%C4%8Dka
[5] https://www.spoj.com/
[6] https://www.ksp.sk/ulohy/zadania/1240/

Formát vstupu

V prvom riadku je číslo $n (1 \leq n \leq 2\,000)$ – veľkosť tabule. Nasleduje n riadkov, každý presne n znakov dlhých – obrázok tabule. Znak `'#'` predstavuje popísané políčko, znak `'.'` nepopísané políčko.

Formát výstupu

Vypíšte jediné číslo – počet bonusových bodov, ktoré Roman vie získať, a znak nového riadku.

Príklad

vstup	výstup
3 .#. ### .#.	1

Takto vyzerá nepočmárané plus veľkosti 1. Romanko zaň dostane 1 bonusový bod.

vstup	výstup
8 ...#.... ...#.... ...#.... ######## ...#.... ...#..#. ...#.### ...#..#.	3

V pravom dolnom rohu je nepočmárané plus veľkosti 1. Štvorec s ľavým horným rohom na druhom políčku druhého riadku a s pravým dolným rohom na 6. políčku 6. riadku je nepočmárane plus veľkosti 2. Romanko teda vie získať najviac $1 + 2 = 3$ *body.*

Štvorec s rohmi na políčkach $(1, 1)$ *a* $(7, 7)$ *je počmárané plus, keďže v pravom dolnom rohu mu zavadzajú popísané políčka menšieho pluska.*

6. O introvertných a extrovertných holuboch

kat. O; 10 b za popis, 10 b za program

POZOR: táto úloha má neštandardný spôsob hodnotenia. Viac sa dozviete v časti Hodnotenie.

Pán Dirichlet je veľkochovateľ poštových holubov. Vo svojich početných holubníkoch chová obrovské množstvo holubov, z ktorých občas niektoré predá malochovateľom.

Dávid sa chce stať malochovateľom poštových holubov. Na záhrade si postavil celý rad holubníkov, ktoré (v duchu najlepších chovateľských tradícií) očísloval číslami $1, 2, 3, \ldots$. Následne si u pána Dirichleta objednal n holubov.

Pán Dirichlet teda Dávidovi pekne po jednom poslal[7] n holubov.

Holuby sú teraz na ceste a Dávid sa pripravuje na ich prijatie. Plán je jednoduchý: holuby budú po jednom prilietať (v poradí, v akom boli poslané, keďže, ako je všeobecne známe, slušné poštové holuby sa navzájom nepredbiehajú) a vždy, keď nejaký holub priletí, Dávid ho strčí do niektorého z holubníkov. To však nemôže robiť len tak halabala. Sťahovanie sa z jedného holubníka do druhého je totiž pre takého holuba veľká udalosť a ak sa to urobí zle, holub si nikdy na svoj nový domov nezvykne, začne chradnúť a predčasne umrie. To, či sa holub v novom domove dobre aklimatizuje, sa riadi nasledovnými pravidlami:

- Každý holub má nejakú *extroverciu*. Extrovercia holuba je nezáporné celé číslo, ktoré je pri introvertných holuboch nízke a pri extrovertných holuboch vysoké.
- Keď holuba s extroverciou e nasťahujeme do nového holubníka, v ktorom už býva **presne** e iných holubov, náš holub sa do tejto spoločnosti bez problémov začlení a rýchlo (hneď) si na nový domov zvykne. Ak je však v holubníku iný počet holubov ako e (menej, alebo viac), náš holub sa v kolektíve necíti dobre a na nový domov si nikdy nezvykne.
- Keď si už raz holub zvykne na svoj holubník (čo sa stane buď hneď po jeho príchode, alebo nikdy), nevadí mu, ak do tohto holubníka pridávame aj nové holuby.

[7] taký poštový holub sa posiela veľmi jednoducho: poviete mu, na akú adresu má ísť a on tam doletí po vlastných

Pán Dirichlet, ako dobrý chovateľ, pozná extrovercie všetkých svojich holubov a spolu s faktúrou poslal[8] Dávidovi aj zoznam extrovercií odoslaných holubov, v poradí, v akom ich poslal. Dávid sa teraz na tento zoznam pozerá a snaží sa naplánovať, ako holuby napchať do holubníkov tak, aby sa všetky dobre aklimatizovali. Pomôžte mu!

Úloha

Na vstupe dostanete postupnosť n čísel $e_1, e_2, \ldots, e_n$ – extrovercie holubov v poradí, v akom budú prilietať. Nájdite takú postupnosť prirodzených čísel $h_1, h_2, \ldots, h_n$, že ak Dávid strčí prvého holuba do holubníka číslo h_1, druhého do holubníka číslo h_2, tretieho do h_3 atď., všetky holuby si dobre zvyknú. Inými slovami, pre každé $i \in \{1, 2, \ldots, n\}$ musí platiť, že presne e_i spomedzi čísel $h_1, h_2, \ldots, h_{i-1}$ sa rovná číslu h_i. Môžete predpokladať, že Dávid má aspoň n holubníkov a aspoň jedna vhodná postupnosť existuje (pán Dirichlet je dobrý chovateľ a neposlal by svoje holuby tak, aby boli niektoré z nich odsúdené na chradnutie).

Formát vstupu

V prvom riadku vstupu je celé číslo n ($1 \leq n \leq 1\,000\,000$). V druhom riadku je n medzerami oddelených nezáporných celých čísel $e_1, e_2, \ldots, e_n$ – extrovercie holubov. Pre každé zmysluplné i platí $0 \leq e_i \leq n - 1$.

Formát výstupu

Vypíšte jeden riadok a v ňom n celých čísel oddelených medzerami – jednu vhodnú postupnosť $h_1, h_2, \ldots, h_n$. Pre každé zmysluplné i musí platiť $1 \leq h_i \leq n$ (aj tak určite nebudete potrebovať viac ako n holubníkov). Ak je možných riešení viac, vypíšte ľubovoľné z nich.

Hodnotenie

V tejto úlohe budeme primárne hodnotiť **pamäťovú zložitosť** vášho algoritmu a až sekundárne časovú. Na časovú zložitosť budeme prihliadať iba tak, ako v iných úlohách prihliadame na pamäťovú (teda algoritmus s lepšou pamäťovou a horšou časovou zložitosťou považujeme za lepší, pokiaľ nie je jeho časová zložitosť absurdne veľká). Tomu zodpovedá aj pomerne voľný časový limit a veľmi malý pamäťový limit (najväčšie vstupy sa vám nezmestia do pamäte) v praktickom testovaní.

Pre jednotlivé vstupné sady platí:

číslo sady	1	2	3, 4, 5
maximálne n	10	1000	1,000,000

V sade č. 3 navyše v každom vstupe platí, že existuje riešenie využívajúce nanajvýš 1 000 holubníkov.

Príklad

vstup

```
4
0 0 0 0
```

výstup

```
1 2 3 4
```

Každý holub musí ísť do nového holubníka. Dobrým riešením je napríklad aj 4 3 2 1.

vstup

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

výstup

```
3 3 3 3 3
```

Všetky holuby musíme dať do rovnakého holubníka.

vstup

```
8
0 0 1 0 1 2 3 2
```

výstup

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

Jedno z mnohých možných riešení.

7. Olympiáda vo vyhľadávaní v texte

kat. O; 12 b za popis, 8 b za program

V Absurdistane sa tento rok koná historicky prvý ročník Olympiády vo vyhľadávaní v texte. Súťaže sa

[8] mailom, nie holubou poštou, takže to Dávidovi príde skôr, ako holuby

zúčastňujú trojčlenné tímy, a koná sa niekoľko zápasov – v každom zápase proti sebe nastúpia dva tímy. Porazený tím súťaž opúšťa.

Každý zápas má nasledovnú formu:

- Každý tím ukáže protivníkovi zdrojový kód programu, ktorým bude vyhľadávať v texte.
- Oba tímy si tajne vyberú text, v ktorom bude ich súper vyhľadávať. Tiež zvolia slovo, ktoré chcú, aby ich súper našiel v texte. Samozrejme, je nejaké obmedzenie na veľkosť vstupu.
- Spustia sa programy oboch tímov na vstupe, ktorý zadal protivník. Úlohou programov je nájsť všetky výskyty zadaného slova v zadanom texte. Vyhráva rýchlejší z programov.

Knuth, Morris a Pratt sa tiež zaregistrovali na súťaž so svojím algoritmom[9]. Práve začal ich prvý zápas, a ich súperom je niekto, kto je tak naivný, že do súťaže prišiel s naivným algoritmom na vyhľadávanie v texte:

```
text[1..n] // text dlzky n, v ktorom vyhladavame
slovo[1..m] // slovo dlzky m, ktore hladame v texte
counter = 0

// postupne skusime kazdu poziciu textu ako zaciatok slova
for i = 1 to n:
  // zlava doprava overime, ci sa tam nachadza slovo
  for j = 1 to m:
    if i + j - 1 > n:
      break
    counter += 1
    if text[i + j - 1] != slovo[j]:
      // lisi sa v j-tom znaku, slovo urcite nezacina na pozicii i v texte
      break
    if j == m:
      // nasli sme vyskyt slova v texte, podame o tom oznam a pokracujeme dalej vo vyhladavani
```

Chcú protivníka totálne zničiť. Vymysleli niekoľko rôznych vstupov, no teraz sa nevedia zhodnúť, ktorý z nich by mali dať protivníkovi. Pomôžte im zistiť, ktorý vstup bude pre protivníka najhorší.

Úloha

Zadaný je text, v ktorom vyhľadávame, a hľadané slovo. Zistite, koľko porovnaní spraví na tomto vstupe naivný algoritmus. (Zaujíma nás teda hodnota premennej `counter` po skončení algoritmu.)

Formát vstupu

Na prvom riadku vstupu je text, v ktorom vyhľadávame. Na druhom riadku je hľadané slovo. Obe slová obsahujú iba malé písmená anglickej abecedy, a majú dĺžku aspoň 1 a najviac 1 000 000 znakov.

Formát výstupu

Na jediný riadok výstupu vypíšte jedno číslo – počet porovnaní, ktoré spraví naivný algoritmus na vstupe. Toto číslo môže byť veľké, a nemusí sa zmestiť do 32-bitovej premennej (`int` v `c++`), odporúčame preto použiť 64-bitové premenné (`long long`).

Príklady

vstup	výstup
aaaa aab	9

8. Obrana pobrežia

kat. O; 10 b za popis, 10 b za program

Kiribatské pobrežie začali nedávno sužovať nájazdy útočných sépií. Sépie priplávajú ku brehu a potom sťahujú do mora slnečníky, lehátka a výletníkov. Minule dokonca odtiahli aj stánok so zmrzlinou.

[9] https://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm

Určite netreba vysvetľovať, čo by sa stalo s turistickým ruchom, ak by sa to rozkríklo. Kiribatská pobrežná stráž preto rýchlo dostala z príslušného ministerstva dotáciu a dôraznú inštrukciu, nech problém rýchlo vyrieši.

Za peniaze z ministerstva pobrežná stráž nakúpila n balíst. Balista je mechanické zariadenie, schopné strieľať harpúny po sépiách. Harpúna má samozrejme na sebe uviazané lano, takže trafenú sépiu vie pobrežná stráž vytiahnuť na breh (a tam naporcovať a zjesť).

Úloha

Pobrežie Kiribati si predstavíme ako os x, more bude polrovina zodpovedajúca kladnému y. Na pobreží stojí n balíst. Z mora sa práve blíži n sépií. Každú balistu aj každú sépiu považujeme za bod.

Pobrežná stráž by chcela uloviť všetkých n sépií naraz. Je zjavné, že na to treba každou balistou uloviť inú sépiu. Existuje teda $n!$ možností, ako priradiť balisty ku sépiám.

Nie všetky možnosti sú však vhodné. Problém je, že keby sme strieľali len tak hala-bala, mohli by sa nám niektoré laná z harpún o seba zamotať. Pobrežná stráž preto musí strieľať takým spôsobom, aby sa tejto nepríjemnosti zaručene vyhla.

Presnejšie: Keď trafíme sépiu harpúnou, lano uviazané o harpúnu tvorí úsečku spájajúcu balistu na brehu s trafenou sépiou vo vode. Priradenie balíst sépiám je *dobré* vtedy, ak sú všetky tieto úsečky navzájom disjunktné (čiže ak žiadne dve nemajú spoločný bod).

Dané sú súradnice balíst aj sépií. Spočítajte, koľko existuje dobrých priradení balíst sépiám. Vypíšte túto hodnotu modulo $10^9 + 7$.

Dôležité: V tejto úlohe nie je nutné optimalizovať časovú zložitosť. Presnejšie, ľubovoľné riešenie s polynomiálnou časovou zložitosťou môže dostať plný počet bodov za popis – ale samozrejme, na plný počet bodov musíte riešenie dobre popísať a ukázať o ňom, že naozaj má polynomiálnu časovú zložitosť.

Formát vstupu

V prvom riadku vstupu je celé číslo n. Platí $1 \leq n \leq 50$.

V druhom riadku vstupu je n celých čísel $b_1, \ldots, b_n$: x-ové súradnice balíst. Balista i sa teda nachádza na súradniciach $(b_i, 0)$.

V treťom riadku je n celých čísel $x_1, \ldots, x_n$: x-ové súradnice sépií.

Vo štvrtom riadku je n celých čísel $y_1, \ldots, y_n$: y-ové súradnice sépií.

Všetky súradnice sú od -1000 do 1000, vrátane. Všetky y_i sú navyše kladné. Žiadne dve balisty ani žiadne dve sépie sa nenachádzajú v tom istom bode.

Formát výstupu

Vypíšte jeden riadok a v ňom hodnotu ($p \bmod 10^9 + 7$), kde p je počet dobrých priradení balíst sépiám.

Príklady

vstup

```
2
4 -4
1 2
1 2
```

výstup

```
2
```

Balisty sú na súradniciach $(4, 0)$ *a* $(-4, 0)$. *Sépie sú na súradniciach* $(1, 1)$ *a* $(2, 2)$. *Sú dve možné priradenia balíst sépiám. Obe sú dobré.*

vstup

```
2
4 -4
4 -4
10 10
```

výstup

```
1
```

Tentokrát je len jedno priradenie balíst sépiám dobré, pri druhom sa laná prekrížia.

vstup

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

výstup

```
2
```

Všimnite si, že sú zakázané aj situácie, v ktorých koniec jednej úsečky leží na inej úsečke. (Inými slovami, lano ide priamo ponad inú sépiu.)

vstup

```
4
-2 2 98 102
0 1 100 101
1 2 1 2
```

výstup

```
4
```

vstup

```
9
5 7 8 9 10 11 12 15 18
13 4 1 6 12 7 9 17 5
15 7 11 2 15 1 6 4 5
```

výstup

```
10
```

Zadania kategórie T

Nezabudnite, že môžete riešiť aj kategóriu T (je trocha ťažšia ako kategória O, ale mnohí z vás ju určite zvládnu).

Body z tejto kategórie sa mierne zohľadňujú aj pri výbere tímu, ktorý pôjde súťažiť na Medzinárodnú olympiádu v informatike (IOI), tento rok v Iráne.