



Korešpondenčný seminár z programovania

Leták zimnej časti XXXIV. ročníka

Korešpondenčný seminár z programovania (KSP) je súťaž programátorov – stredoškôľakov 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 série príkladov.
- 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.
Úlohy rieš samostatne a neprezrádzaj riešenia ostatným riešiteľom. Opisovanie riešení a prezradenie riešení pred termínom série je porušením pravidiel KSP. Po skončení série sa samozrejme o riešeniach rozprávať môžeš. :)
- **Odovzdaj** riešenia úloh. Odkaz na odovzdávanie úloh nájdeš webovým zadaním každej úlohy alebo na stránke <https://www.ksp.sk/odovzdavanie>. Na odovzdávanie sa treba prihlásiť na stánku, 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/á s výsledkom. Ak nevieš, ako majú vyzeráť odovzdané programy, pozri si <https://www.ksp.sk/idealny-program>
 - 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/idealne-riesenie>. Popis opravuje a obodujú vedúci KSP po skončení série.
- Po skončení série 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.
- Víťazi všetkých kategórií (to sú bežne prví traja riešitelia v oficiálnej koncovej výsledkovej listine letnej aj zimnej časti) od nás dostanú **vecné ceny** v celkovej hodnote aspoň 60 Eur. Ceny si budú môcť vybrať z viacerých typov – knižka, spoločenská hra, hlavolam, počítačová hra. . .
- Pre najlepších 32 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 kategórií 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). Úspešní riešitelia tiež dostanú pekné **diplomy**.
- Netreba zabunúť ani na to, že riešením súťažných úloh a štúdiom našich vzorových riešení sa toho môžeš veľa naučiť. Získané poznatky a skúsenosti sa ti iste budú hodiť v iných súťažiach v programovaní (napríklad pri riešení **Olympiády v informatike**), v bežnom živote aj **počas vysokoškolského štúdia**. Mnoho našich bývalých riešiteľov sa bez ťažkostí zamestnalo v špičkových IT spoločnostiach ako **Google**, **Facebook**, **ESET**, ... a už ako vysokoškoláci sa dostávajú na stáže v týchto firmách, napríklad vo Švajčiarsku alebo v Amerike.

Kategórie a riešenie

- Na riešenie úloh série 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.
- KSP má tri kategórie úloh. Z, O a T.
 - **Kategória Z** je pre začiatočníkov a patria do nej najľahšie úlohy s číslami 1 až 5. Po tom, ako budete niekoľkokrát úspešní riešitelia kategórie Z alebo zažijete 2 sústredenia, už rozhodne nie ste začiatočníci. Neoficiálne môžete túto kategóriu riešiť koľkokrát chcete, ale najviac dvakrát môžete byť oficiálni úspešní riešitelia (a dostať diplom či vyhrať ceny) a najviac dvakrát (nezávisle od toho koľkokrát ste boli úspešní) budete pozvaní na sústredenie. Následne sa treba posunúť ďalej a riešiť kategóriu O.
 - **Kategória O** je pre všetkých riešiteľov a patria do nej úlohy s číslami 4 až 8. Môžete ju riešiť koľkokrát chcete, kým ste na strednej (alebo základnej) škole.
 - **Kategória T** je pre pokročilejších riešiteľov, alebo pre riešiteľov, ktorým sa zdá, že je úloh málo. Obsahuje samostatných 3 až 5 úloh, ktoré sa zverejňujú trochu neskôr ako úlohy prvých kategórií. V kategórii T sa bežne neodovzdáva popis, hodnotí sa len program. Určite neváhajte, pozrite si tieto príklady a skúste ich vyriešiť, najmä ak už ste aspoň raz riešili kategóriu O. Podobne, ako v kategórii O, všetci stredoškooláci sú vítaní pri riešení týchto úloh.
- Medzi úlohami sa nachádzajú ľahšie aj ťažšie. Väčšinou tie s menším číslom sú ľahšie, ale obtiažnosť príkladu je subjektívna. Preto odporúčame prečítať si zadania všetkých úloh a zamyslieť sa nad všetkými. Každý si môže vybrať tie úlohy, ktoré vie riešiť a ktoré považuje za zaujímavé. A nezabudnite, že aj za čiastočné riešenia sú nejaké body.

Registrácia

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

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



Úlohy 1. kola zimnej časti

Termín odoslania riešení tejto série je pondelok 24. októbra 2016.

1. Zavesme tam niečo!

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

Izby na internátoch nie sú veľmi zaujímavé. Všetky vyzerajú rovnako – na jednej stene sú skrine, na druhej okno a balkón, zvyšné dve sú skoro prázdne. Niektorí študenti si na to jednoducho zvyknú a často chodia domov, tí ostatní sa snažia svoje izby trochu zútulniť.

Baška a Olívia sa o to snažili tiež. Doniesli si farebné obliečky na paplóny, kúpili si koberec, oblepili ošarpané poličky pekným papierom a dokonca aj vymaľovali. Keď už boli takmer spokojné, všimli si veľkú prázdnu plochu na stene nad poličkou. Hneď im bolo jasné, že tam treba niečo zavesiť. Zhodli sa, že to bude obraz a že by mal byť veľký, aby zakryl čo najviac z ich steny. Vedeli, akú plochu má ich stena, stačilo už len nájsť k nej dostatočne vhodný obraz.

Vybrali sa preto do obchodu. Obchod bol veľký a bolo v ňom veľa stien a veľa obrazov. Nevedeli sa na žiadnom zhodnúť, tak si ich všetky nafotili, aby si ich mohli v pokoji na izbe pozrieť a vybrať ten najvhodnejší.

Úloha

Na vstupe dostanete jednu fotku v podobe [ASCII art](#). Vašou úlohou je nájsť na fotografii obraz a zistiť, aký je veľký. Obraz sa skladá z ľubovoľných znakov a medzier. Je celý ohraničený obdĺžnikovým rámom. Zvyšok vstupu tvoria medzery a znázorňujú stenu okolo obrazu na fotografii. Veľkosť obrazu je plocha steny, ktorú zakryje, teda jeho výška krát šírka v ASCII znakov (každý obraz je obdĺžnikový).

Formát vstupu

Na vstupe je niekoľko rovnako dlhých riadkov ukončených znakom konca riadku. Dĺžka jedného riadku neprekročí 250 znakov. Rám sa skladá zo znakov + v rohoch, - navrchu a dole a znaku | na bokoch. Každý obraz má veľkosť minimálne 1 (do tejto hodnoty sa neráta plocha rámu).

Formát výstupu

Vypíšte jedno číslo – veľkosť plochy obrazu bez rámu. Nezabudnite za výsledkom vypísať koniec riadku.

Príklad

vstup

```
+-----+
|   .- .- .- .- .-   |
|  | / / { { _ | { } } |
|  | \ \ .- .- } | .-' |
|  ' _ ' _ ' _ ' _ ' |
+-----+
```

výstup

108

Obraz má šírku 27 znakov a výšku 4 znaky, t.j. veľkosť $27 \times 4 = 108$. Na začiatku sú 4 riadky plné medzier, pred obrazom sú na každom riadku 3 medzery a za ním 4, na konci je ešte 5 riadkov plných medzier. (Vstup z pdf zadania sa vám možno neskopíruje v správnom formáte)

2. Zrýchlenie parkoviska

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

Niektorí riešitelia KSP sa po maturite rozutekajú do sveta. Napríklad do Dánska. Nechcú si ale so sebou brať svoje autá a keďže sú dobré duše, požičajú ich Trojstenu.

Toto má ale nežiadúce vedľajšie efekty: matfyzácke parkovisko sa rýchlo plní a bezradní šoféri sa hádajú o prázdne miesta. Preto sa vedenie rozhodlo zaviesť nový systém pridelovania parkovacích miest, ktorý bude férový ku všetkým.

Úloha

Máme parkovisko, kde je m parkovacích miest, očíslovaných číslami od 0 po $m - 1$.

Na parkovisko prichádzajú a odchádzajú autá s unikátnou ŠPZ (teda na parkovisku nebudú naraz dve autá s rovnakou ŠPZ, ale jedno auto môže prísť a odísť aj viackrát).

Vašou úlohou je pridelovať autám parkovacie miesta. Každému autu, ktoré príde, pridelíte ľubovoľné voľné parkovacie miesto. Ak sú všetky miesta plné, miesto pridelovať nemusíte a auto neobslúžite (ide parkovať inam).

Formát vstupu

V prvom riadku vstupu sú čísla m ($1 \leq m \leq 10^5$) a n ($1 \leq n \leq 10^6$). Číslo m je počet miest na parkovisku a n je počet príchodov a odchodov áut dokopy.

V každom z ďalších n riadkov je číslo ŠPZ ($1 \leq SPZ \leq 10^6$) auta, ktoré prišlo alebo odišlo. Ak sa auto s touto ŠPZ nachádza na parkovisku, práve odišlo. V opačnom prípade práve prichádza.

Formát výstupu

Pre každé auto, ktoré prišlo alebo odišlo, vypíšete jeden riadok a na ňom číslo parkovacieho miesta, ktoré obsadilo alebo uvoľnilo (číslované od 0 po $m - 1$). Ak príde auto na plné parkovisko, vypíšete jeden riadok a na ňom reťazec „plne“.

Správny výstup je ľubovoľné pridelovanie miest, pri ktorom žiadnemu autu nebude pridelené už obsadené miesto a každé auto pri odchode uvoľnilo miesto, ktoré mu bolo pridelené.

Príklad

vstup

```
3 7
4
9039
103
19
4
103
47
```

výstup

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

3. Znova začíname

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

Znova je to tu. Začal nám nový ročník KSP a s tým spojené prvé kolo. Zahrievajú sa servery, vedúci si pripravujú červené perá vo svojich obľúbených softvéroch na editovanie PDF a distribúcia zadaní je v plnom prúde. Začal pracovať aj náš tím odvážnych webových vývojárov. Zhodli sa, že v novom roku treba softvér aj testovať.

Zaručene najdôležitejšou časťou celej série je výsledkovka. Zodpovední vývojári preto chcú pripraviť test, ktorý overí, či daná vygenerovaná [výsledkovka](#) mohla vzniknúť korektným spôsobom. Zvládnete im pomôcť?

Úloha

Zadanú dostanete *výsledkovku* – zoznam mien riešiteľov od najlepšieho po najhoršieho. Z výsledkovky sa pre každého riešiteľa dozvieme aj to, ako sa jeho umiestnenie zmenilo oproti minulej sérii:

- *Zlepšenie sa* znamená, že poradie účastníka v aktuálnej výsledkovke je lepšie (menšie) ako jeho poradie v predošlej výsledkovke.
- *Zhoršenie sa* znamená, že poradie účastníka v aktuálnej výsledkovke je horšie (väčšie) ako jeho poradie v predošlej výsledkovke.

- Za predpokladu, že sa poradie nezväčšilo, ani nezmenšilo, hovoríme, že sa jeho poradie *zachovalo*.

Zistite, či zadaná výsledkovka mohla vzniknúť korektným spôsobom z nejakej predošlej výsledkovky a jednu takúto predošlú výsledkovku nájdite. Z dôvodu zachovania jednoduchosti zadania rátame s tým, že účastníci medzi sériami nepríbúdajú, nemiznú a nemôžu byť dvaja na tom istom mieste.

Formát vstupu

Z prvého riadku načítajte z ($1 \leq z \leq 100\,000$) – počet riešiteľov.

z nasledujúcich riadkov zobrazuje aktuálne výsledky zoradené od najlepšieho riešiteľa (najmenšie umiestnenie) po najhoršieho. Každý riadok začína znakom, ktorý určuje relatívny posun voči predošlej výsledkovke: „+“ pre zlepšenie, „-“ pre zhoršenie, „0“ pre zachovanie poradia. Za medzerou nasleduje meno riešiteľa pozostávajúce z malých písmen anglickej abecedy (dĺžky najviac 20).

Formát výstupu

Na prvý riadok vypíšte reťazec „OK“, ak daná výsledkovka mohla vzniknúť korektné. Na ďalších z riadkov vypíšte jedno možné predošlé poradie riešiteľov (od najlepšieho po najhoršieho).

Vypíšte reťazec „FAIL“, ak je daná výsledkovka určite chybná.

Príklad

vstup	výstup
3 + peter - alica 0 bob	OK alica peter bob
vstup	výstup
1 - peter	FAIL

4. Zápolenie o cukríky

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

Jedno pomalé štvrtkové popoludnie Kuko zase raz dostal chuť na cukríky. V tú ranu vošiel do miestnosti Mišo s jeho obľúbeným balením všetkých chutí. Kukovi v hlave vzkrsol nápad – staval sa s Mišom o cukrík, že proti nemu vyhrá partiu hry Go. Tak hrali jednu hru, dve hry, až napokon za svitu posledných líčov zapadajúceho slnka vyprázdni celú sáčok.

Kukovi sa podarilo vyhrať každú jednu hru. Teraz, keď už nemá žiadne ďalšie cukríky na zjedenie, začal uvažovať, či by mu celé balenie nechutilo viac, ak by niektoré hry nechal vyhrať Miša.

Kukove chuťové poháriky totiž fungujú prazvláštne. Ak si dá len jeden cukrík, je mu jedno aký bude, všetky mu budú chutiť rovnako. Ak ale zje dva rovnaké po sebe, situácia je trochu zložitejšia. Niektorých chutí by sa nevedel nabažiť ani keby ste mu doviezli plný kamión, pri iných by druhý rovnaký cukrík zjedol len pri použití násilia.

Teraz by od vás chcel vedieť, ako najviac mu mohlo Mišove balenie chutiť, ak by niektoré hry nevyhral.

Úloha

Na vstupe máte postupnosť cukríkov dĺžky n , o ktoré postupne Kuko s Mišom bojovali. Pre každý typ cukríka i , tiež viete, akú má chutnosť, ak je zjedený po cukríku rovnakého typu – túto chutnosť nazveme b_i . Tiež viete, akú má chutnosť cukrík, ak je zjedený po cukríku iného typu – chutnosť c . Táto chutnosť je rovnaká pre všetky typy cukríkov.

Prvý cukrík má chutnosť rovnakú, ako keby bol zjedený po cukríku iného typu (teda c), chutnosť každého ďalšieho cukríka závisí na type posledného zjedeného cukríka.

Nájdite chutnosť najchutnejšej podpostupnosti cukríkov (nie nutne súvislej). Chutnosť podpostupnosti je rovná súčtu chutností jednotlivých zjedených cukríkov.

Formát vstupu

V prvom riadku vstupu sú kladné, medzerou oddelené čísla n , udávajúce počet cukríkov, a m , udávajúce počet typov cukríkov. Platí, že $1 \leq m \leq n \leq 1\,000\,000$

V nasledujúcom riadku sa nachádza n medzerou oddelených čísel a_i ($1 \leq a_i \leq m$) – typ i -teho cukríka.

V ďalšom riadku nasleduje m medzerou oddelených čísel b_i ($0 \leq b_i \leq 10^9$) – chutnosť cukríka typu i , ak bol zjedený hneď po cukríku rovnakého typu.

V poslednom riadku je jedno kladné číslo c ($0 \leq c \leq 10^9$) – chutnosť cukríka, ak bol zjedený hneď po cukríku iného typu, alebo je to prvý zjedený cukrík.

Všimnite si, že súčet chutností cukríkov ľahko presiahne $2^{31} - 1 \approx 2 \cdot 10^9$, čo je najväčšie číslo, ktoré sa dá uložiť v 32-bitovej premennej so znamienkom. Použite preto 64-bitové premenné typu `long long` v C++ a `Int64` v Pascale.

Formát výstupu

Na jediný riadok vypíšete chutnosť najchutnejšej podpostupnosti cukríkov, akú vie Kuko dosiahnuť.

Príklad

vstup	výstup
<pre>5 3 1 2 3 3 2 3 7 1 2</pre>	<pre>11</pre>

Najchutnejšia podpostupnosť je 1, 2, 2, pričom jednotlivé chutnosti cukríkov sú 2, 2, 7. Prvý a druhý cukrík majú obidva chutnosť $c = 2$, keďže pred nimi nebol zjedený cukrík rovnakého typu. Všimnite si tiež, že ak by Kuko zjedol všetky cukríky, celková chutnosť by bola menšia, konkrétne $2 + 2 + 2 + 1 + 2 = 9$.

5. Oblepené mesto

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

Bratislava, hlavné mesto Slovenska, je celkom slušne oblepená. To určite viete, ak ste v nej aspoň raz v živote boli. Billboard sem, billboard tam, plagáty a plagáty. Ale ani jeden nepatrí spoločnosti Kaderníkov Spod Papuče. Jej manažér, Mário, si to tiež všimol a tak sa rozhodol, že jeden plagát svojej spoločnosti predsa len vybaví.

Pochodil celú Bratislavu a nakoniec našiel ideálnu stenu. Bolo na nej už síce zopár plagátov, ale to nevadí. “Veď nejaké môžeme aj prelepiť,” povedal si Mário. V Bratislave je však zakázané prelepiť nejaký plagát čiastočne, a tak je to aj správne, veď je to potom hrozne neestetické. Ak sa má nejaký plagát prelepiť, tak celý!

Teraz Mário nevie, ako má nalepiť plagát Kaderníkov Spod Papuče. Pomôžete mu?

Úloha

Na stene je už umiestnených niekoľko plagátov, ktoré majú tvar obdĺžnikov, ktorých strany sú rovnobežné s osami súradnicovej sústavy. Mário si zaumienil, že plagát svojej spoločnosti musí pokrývať aspoň obdĺžnik (x, y, w, h) (x, y určujú pozíciu ľavého horného rohu, w, h určujú šírku a výšku).

Teraz potrebuje nájsť takú obdĺžnikovú plochu na nalepenie plagátu, aby pokrývala aspoň obdĺžnik (x, y, w, h) a zároveň, aby každý iný plagát prekrývala buď celý alebo vôbec.

Keďže spoločnosť Kaderníkov Spod Papuče má len obmedzené financie, Mário chce nájsť najmenší takýto obdĺžnik.

Formát vstupu

Na prvom riadku sa nachádza jedno celé číslo n ($0 \leq n \leq 100\,000$) – počet plagátov na stene.

Na ďalších n riadkoch sa nachádzajú popisy jednotlivých plagátov. Na i -tom z týchto riadkov sú štyri celé čísla oddelené jednou medzerou: x_i, y_i, w_i, h_i . Čísla x_i, y_i určujú súradnice ľavého horného rohu i -teho plagátu a čísla w_i, h_i určujú jeho šírku a výšku. Tieto plagáty sa môžu čiastočne aj úplne prekrývať.

Pozor, v tejto úlohe používame súradnicovú sústavu, kde sa y -ová súradnica zväčšuje smerom dole. Preto určujeme pozíciu ľavého horného rohu.

Pre všetky i platí $0 \leq x_i, y_i, w_i, h_i \leq 10^9$.

Na poslednom riadku sa nachádzajú čísla x, y, w, h oddelené medzerou. Tieto popisujú plochu, ktorá musí byť pokrytá Máriovým plagátom.

Rovnako platí, že $0 \leq x, y, w, h \leq 10^9$.

Formát výstupu

Vypíšete jeden riadok a na ňom štyri celé čísla oddelené jednou medzerou: x_p, y_p, w_p, h_p , kde x_p, y_p sú súradnice ľavého horného rohu a w_p, h_p sú šírka a výška plochy kam nakoniec Mário nalepiť plagát svojej spoločnosti.

Príklady

vstup

```
2
0 7 5 5
7 0 5 5
0 0 5 5
```

výstup

```
0 0 5 5
```

Máriova vysnívaná plocha sa neprekrýva so žiadnym existujúcim plagátom.

vstup

```
2
0 5 5 5
5 0 5 5
0 0 5 5
```

výstup

```
0 0 5 5
```

Dokonca ani teraz sa neprekrýva so žiadnym existujúcim plagátom.

vstup

```
2
2 2 9 9
6 0 7 2
1 1 4 4
```

výstup

```
1 0 12 11
```

Cieľová plocha sa prekrýva s prvým plagátom. Ak ho však chceme zahrnúť, nevieme sa vyhnúť prekrytiu s druhým, a tak musíme zakryť aj ten.

6. Obnova bytového parku

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

Nie je tomu tak dávno, čo sa lokálny magnát (volajme ho M) rozhodol odkúpiť od mesta nejaké paneláky a zrekonštruovať byty v nich. Samozrejme, najlepší spôsob, ako s takýmto biznisom začať, je podať si grant. Ako to už však chodí, na získanie grantu potrebujete splniť absurdné požiadavky. Ani tentoraz nedošlo k výnimke a magnát M sa musel popasovať hneď s niekoľkými:

- Počet kúpených panelákov musí byť práve k .
- Paneláky sa musia nachádzať v štvrti, kde sú budovy usporiadané do štvorcovej siete, teda všetky majú rovnako veľký, štvorcový pôdorys.
- Všetky kúpené paneláky spolu musia susediť stenami.
- Čím je stavba nižšia, tým viac je ekologická a teda vhodnejšie vyvažuje charakter mesta.
- Mačka musí mať štyri nohy a nestrká sa do mikrovlnky.

Aj magnátovi M chvíľu trvalo, kým pochopil, o čo sa tu jedná. Zistil, že úradníci si od neho vyúčtujú ekologický poplatok, na základe toho, akú najvyššiu budovu bude rekonštruovať.

Jeho cieľom je teda splniť všetky podmienky a zároveň spraviť čo najvýhodnejší biznis – vybrať si takých k bytoviek, že vytvoria súvislú oblasť a najvyššia budova medzi nimi je čo najnižšia (medzi všetkými takýmito oblasťami).

Úloha

Dostanete číslo k a mriežku čísel s r riadkami a s stĺpcami.

Hovoríme, že oblasť mriežky je súvislá, ak sa vieme z každého jej políčka dostať na každé iné políčko oblasti len prechodmi smermi hore, dole, vpravo, vľavo, bez opustenia danej oblasti.

Nájdite takú súvislú oblasť, ktorá obsahuje k čísel a najväčšie číslo, ktoré obsahuje, je najmenšie možné.

Formát vstupu

Na prvom riadku sa nachádza číslo k . Na druhom riadku sú medzerou oddelené čísla r, s . Pre všetky vstupy platí $1 \leq k \leq r \cdot s \leq 10^6$.

Na každom z nasledujúcich r riadkov sa nachádza s medzerou oddelených kladných celých čísel, ktoré udávajú výšky jednotlivých budov v meste. Výška žiadnej budovy nepresiahne 10^9 .

Formát výstupu

Na prvý riadok výstupu vypíšete celé číslo určujúce najmenšiu možnú výšku najvyššej odkúpenej budovy. Následne vypíšete r riadkov a na každom z nich s znakov. Každý znak musí byť bodka ("."), alebo hviezdička ("*"), pričom hviezdičkami označené políčka musia vyznačovať ľubovoľné zo správnych riešení.

Príklady

vstup

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

výstup

```
3
...
..*
.**
***
```

vstup

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

výstup

```
2
***
.*
...
```

7. Osamelý a jednobunkový

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

Bol raz jeden jednobunkový organizmus menom Cecil¹. Bolo mu smutno, lebo nebolo nikoho, s kým by sa mohol porozprávať pri jedle, a tak sa Cecil začal deliť, aby mal koho pozvať na svoje párty.

Časom mal Cecil dosť potomkov, a tak sa rozhodol zorganizovať párty. Pozval na ňu všetkých svojich k -potomkov. (1-potomok Cecila je každé jeho dieťa, 2-potomok je každé jeho vnúča, atď.)

Postupom času začali aj potomkovia Cecila organizovať párty podobným spôsobom. Každému svojmu k -potomkovi pošlú pozvánku na svoju párty. Tá má nasledovnú formu: "Na jeho veľkú párty ťa pozýva tvoj k -ty predok, sám magnificentný..."

Cecilovi potomkovia sú ale inteligentní, lebo to bolo v Cecilových génoch. Je im jasné, že množstvo jedla a kofoly, ktoré sa im na párty ujde, je určené počtom účastníkov párty. Nie vždy sa teda oplatí ísť na párty. Pomôžte im pri rozhodovaní – pre každú pozvánku určite, koľko účastníkov bude na párty ak sa jej všetci pozvaní zúčastnia.

Úloha

Zadaný je zakorenený strom s n vrcholmi. Prichádza vám postupne m otázok. Každá otázka je určená vrcholom v a poradovým číslom predka k . Pre každú otázku zistíte, koľko k -potomkov má k -predok vrchola v . Ak k -predok vrcholu v neexistuje, tak je odpoveď 0.

1-potomok vrcholu v je ľubovoľné jeho dieťa. Pre $k > 1$ je k -potomok vrcholu v ľubovoľné dieťa niektorého $(k - 1)$ -potomka v . 1-predok vrcholu v je otec v ak existuje. Pre $k > 1$ je k -predok vrcholu v otec $(k - 1)$ -predka v .

Úlohu je nutné riešiť online – váš program sa nedozvie ďalšiu otázku, kým neodpovie na tú aktuálnu. Za každou odpoveďou preto použijete `flush(stdio)` alebo `cout << flush`, aby sa výstup vášho programu hneď odoslal testovaču.

Formát vstupu

Na prvom riadku vstupu je celé číslo t ($1 \leq t \leq 1\,000$) – počet testov. Každý z testov má nasledovný formát:

Prvý riadok testu obsahuje jedno celé číslo n ($1 \leq n \leq 100\,000$) – počet vrcholov stromu. Druhý riadok testu obsahuje n čísel p_1, p_2, \dots, p_n – i -te z nich je číslo priameho predka vrcholu i . Ak je vrchol i koreňom stromu, tak $p_i = 0$. (Pre i také, že vrchol i nie je koreňom stromu platí $1 \leq p_i \leq n$.) Môžete predpokladať, že zadaný graf je naozaj strom.

Tretí riadok testu obsahuje jedno celé číslo m ($1 \leq m \leq 100\,000$) – počet otázok. Nasleduje m riadkov, i -ty z nich obsahuje dve celé čísla v_i, k_i ($1 \leq v_i, k_i \leq n$) popisujúce otázku.

Súčet n zo všetkých testov nepresiahne 100 000, a súčet m zo všetkých testov nepresiahne 150 000.

¹Možno ste už počuli o jeho pradedovi Bacilovi.

Formát výstupu

Pre každú otázku vypíšte na samostatný riadok jedno celé číslo – počet k -potomkov k -predka vrcholu v .

Príklad

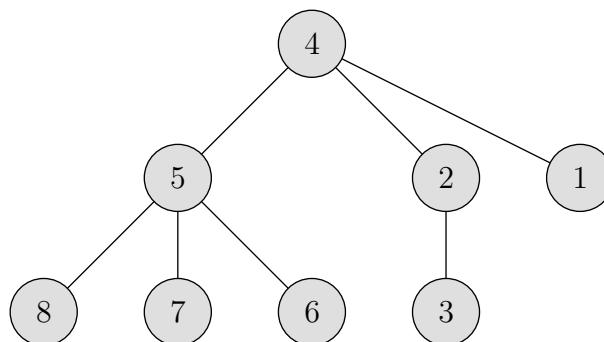
vstup

```
2
8
4 4 2 0 4 5 5 5
6
1 1 // 1-predok 1 je 4.
    // Jeho 1-potomkovia su 5, 2 a 1.
1 2 // 2-predok 1 neexistuje.
3 1 // 1-predok 3 je 2.
    // Jeho 1-potomok je len 3.
3 2 // 2-predok 3 je 4.
    // Jeho 2-potomkovia su 8, 7, 6 a 3.
3 3 // 3-predok 3 neexistuje.
5 1 // 1-predok 5 je 4.
    // Jeho 1-potomkovia su 5, 2 a 1.
5
3 0 2 5 1
5
4 5
4 1
1 2
1 3
2 1
```

výstup

```
3
0
1
4
0
3
0
1
1
0
0
```

Text uvedený za // je komentár – v skutočných vstupoch sa žiadne komentáre samozrejme nachádzať nebudú. Druhý test má správny formát. Strom v prvom teste vyzerá nasledovne:



8. Oprášení startup

kat. O; 0 b za popis, 25 b za program

V dávnych časoch, na ktoré sa v KSP pamätá už len Mišof, sa snažil Dávidko² rozbehnúť start-up so špeciálnymi 'Dávidkovými procesormi'. V tom čase predstavovali so svojou 32 bitovou výpočtovou jednotkou a takmer 77 MB pamäťou vrchol technického pokroku. Bohužiaľ, kvôli slabej mediálnej kampani neboli tieto procesory úspešné a celý Dávidkov projekt zapadol prachom.

Nedávno, prehrabávajúc sa zákutiami T2, objavil tento projekt Sysel a povedal si, že teraz by už mohol dopadnúť lepšie. Ísť však na trh s procesorom by nebolo príliš rozumné. Ak by však daný procesor upravil tak, aby vedel komprimovať text, tak by 'Syslov komprimátor' mohol zožať nečakaný úspech. Akurát tá úprava nie je taká jednoduchá, keďže 'Dávidkov procesor' beží na veľmi starom a primitívnom systéme. Vy si s tým však určite nejak poradíte.

Dávidkov procesor

Dávidkov procesor obsahuje jeden zásobník (stack), s ktorým pracujú dané inštrukcie. Na zásobník sa ukla-

²Bývalý KSP vedúci, ktorý aktuálne pracuje v New Yorku a má manželku Halucinku (ďalšia bývalá KSP vedúca).

dajú 32-bitové čísla so znamienkom, teda čísla z rozsahu -2^{31} až $2^{31} - 1$. Na začiatku behu programu je zásobník prázdny.

Zásobník zapisujeme horizontálne, pričom napravo sa nachádza vrch zásobníka. Reťazec „...“ reprezentuje pôvodný stav zásobníka. Reťazec „...|a|b“ značí, že na vrchu zásobníka sa nachádza číslo b a pod ním číslo a . Hodnota „S[n]“ je n -tá hodnota zásobníka. Ak je $n \geq 0$ táto hodnota sa počíta od spodku, ak $n < 0$, tak od vrchu.

Dávidkov procesor pozná nasledovné príkazy:

Príkaz	Popis príkazu	Zásobník pred	Zásobník po
PUSH <i>const</i>	Na vrch zásobníka pridá číselnú konštantu <i>const</i> const
OUT	Odstráni vrchné číslo zásobníka a vypíše ho na výstup ako ASCII znak. Chyba nastane, ak je odstránené číslo menšie ako 0 alebo väčšie ako 127.	... a	...
READ	Odstráni zo zásobníka vrchné číslo (označme ho a) a následne nakopíruje na vrch zásobníka hodnotu $S[a]$... a	... S[a]
JGZ	Odstráni zo zásobníka dve vrchné hodnoty b a a . Následne sa posunie o b inštrukcií (dopredu pre kladné b , inak dozadu) v prípade, že je $a > 0$... a b	...
ADD	Odstráni dve čísla na vrchu zásobníka a pridá na vrch ich súčet	... a b	... a+b
MUL	Odstráni dve čísla na vrchu zásobníka a pridá na vrch ich súčin	... a b	... a*b
DIV	Odstráni dve čísla na vrchu zásobníka a na vrch pridá ich podiel a zvyšok po delení ($q = a/b$ (celočíselné delenie), $r = a \% b$)	... a b	... q r

Ak by pri vykonávaní niektorého príkazu nebolo v zásobníku dostatočné množstvo čísel (napríklad aspoň 2 pre JGZ, aspoň 1 pre OUT), program skončí chybou. Program tiež skončí chybou, ak je na zásobníku v niektorom momente viac ako 20 000 000 čísel, alebo sa vykoná viac ako 300 000 000 príkazov.

Technické detaily aritmetických operácií (pretečenie, modulo zo záporného čísla atď.) fungujú rovnako ako v C++.

Program si po spustení udržiava hodnotu *príkaz*, ktorá ukazuje na príkaz, ktorý sa aktuálne vykonáva. Táto hodnota je na začiatku 0 (začína sa prvým príkazom). Hodnota *príkaz* sa zvýši o 1 po dokončení aktuálneho príkazu. Ak nastal skok príkazom „JGZ“, táto premenná sa ešte predtým zmení o danú veľkosť skoku b , následne sa však opäť posunie o 1 dopredu. Ak teda $b = -1$, tak *príkaz* bude ukazovať opäť na ten istý príkaz. V okamihu ako premenná *príkaz* ukazuje mimo existujúce príkazy, program je považovaný za korektne zastavený.

Úloha

Dávidkov procesor podporuje iba uvedenú malú sadu príkazov. Vašou úlohou je pomocou týchto príkazov napísať čo najkratší program, ktorý vypíše zadaný text.

Formát vstupu

Dostanete 9 rôznych súborov s textami, všetky používajú štandardnú 7-bitovú ASCII reprezentáciu znakov. Niektoré z textov sú zmysluplné anglické alebo slovenské texty, iné sú vytvorené umelo. Súborov so vstupmi majú názvy 1.in, 2.in, ..., 9.in.

Formát výstupu

Odovzdajte jeden ZIP archív, ktorý bude obsahovať súbory s názvami 1.out, 2.out, ..., 9.out. Pre každý vstupný text teda odovzdávate samostatný súbor s programom pre Dávidkov procesor.

Každý súbor sa musí skladať z príkazov popísaných vyššie a vypísať daný text. Odovzdaný program je

zoznam inštrukcií, každá na samostatnom riadku. Inštrukcie píše veľkými písmenami presne ako sú udané v tabuľke.

Uvedomte si, že vašou úlohou je vypísať presne taký istý text, vrátane znakov ako medzery alebo konce riadkov. Prvé 4 vstupy neobsahujú znak konca riadka.

Hodnotenie

Každý z deviatich textov bude hodnotený samostatne. Za vyriešenie každého z nich sa dá získať nejaká časť bodov a výsledný bodový zisk je súčet bodov za každý text.

Text	1	2	3	4	5	6	7	8	9
Maximálny počet bodov	1	2	2	3	3	3	4	3	4

V tejto úlohe budete súťažiť proti sebe. Bodovanie v rámci jedného textu je škálované podľa najlepšieho odovzdaného riešenia.

Skóre odovzdaného programu sa vypočíta ako počet inštrukcií použitých v programe (teda počet príkazov v zdrojovom kóde, nie počet príkazov, ktoré sa naozaj vykonajú počas behu programu), pričom príkaz „PUSH“ sa počíta za 2. Nech O je skóre programu, ktorý ste odovzdali, M je najmenšie dosiaľ získané skóre nejakého programu (vrátane toho so skóre O) a B je počet bodov, ktoré sa dajú získať za vyriešenie daného textu. Výsledného skóre tohto programu je $B \cdot \frac{M}{O}$.

Počas behu kola budeme hodnoty M (najmenšie dosiahnuté skóre pre každý text) aktualizovať približne raz za týždeň a po skončení kola ich aktualizujeme znova a vypočítame finálne body.

Keďže neodovzdávate programy a nás zaujíma, ako úlohu riešite, môžete do ZIPka priložiť aj zdrojové kódy programov, ktoré ste pri riešení použili, prípadne stručný popis, ak by ste sa chceli pochváliť.

Súbory na stiahnutie

Na tomto odkaze: <http://media.ksp.sk/ulohy/34rocnik/1kolo/prik18-na-stiahnutie.zip> si viete stiahnuť niekoľko užitočných súborov. Je tam spomínaných deväť textov, pre ktoré máte vytvoriť programy a takisto C++ program, ktorý simuluje tento zásobníkový procesor. Vďaka nemu si viete otestovať, či vami vytvorený program naozaj vypíše žiadaný text. Rovnakým spôsobom budeme tiež simulovať vaše riešenia na testovači, takže si môžete overiť implementáciu dôležitých častí zásobníkového procesora.

Príklad

vstup	výstup
Hello	PUSH 72 OUT PUSH 203 PUSH 2 DIV PUSH 2 JGZ PUSH 80 OUT OUT PUSH 108 PUSH 111 PUSH 0 PUSH 0 READ OUT READ OUT OUT

Všimnite si, že príkazy na riadkoch 7 a 8 (číslujeme od 0) boli preskočené vďaka „JGZ“ a nikdy sa nevykonali. Program skončil korektne a na zásobníku ostalo číslo 108. Skóre takéhoto programu je 28.

Zadania kategórie T

Nezabudnite, že môžete riešiť aj kategóriu T (je trochu 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.