

Korešpondenčný seminár z programovania

Leták letnej časti XXVII. ročníka

Tento leták obsahuje zadania prvých kôl Korešpondenčného seminára z programovania kategórií KSP-Z, KSP-O aj KSP-T a pravidiel súťaže. Korešpondenčný seminár z programovania je súťaž v programovaní určená pre žiakov stredných škôl. Ročne sa jej zúčastňuje vyše 200 žiakov. Tento rok pre Vás skupina študentov a doktorandov FMFI UK pripravila už 27. ročník.

Organizácia:

Pre riešiteľov OI organizuje SK OI celoštátny korešpondenčný seminár ako súčasť OI. Zadanie úloh korešpondenčného seminára rozosiela riešiteľom niekoľkokrát do roka. Korešpondenčný seminár pre riešiteľov OI zo svojho kraja prípadne i z ďalších krajov resp. obvodov môžu organizovať aj KK OI. Pre najlepších riešiteľov týchto korešpondenčných seminárov sa obvykle taktiež organizujú sústreďenia.

Slovenský výbor Olympiády v informatike poveruje realizáciou Korešpondenčného seminára z programovania a s ním súvisiacich sústreďení občianske združenie **TROJSTEN**, ktoré zodpovedá za odbornú náplň a program seminára a sústreďení. Pri zabezpečovaní jednotlivých sústreďení bude OZ TROJSTEN spolupracovať s **IUVENTOU**, ktorá pokryje časť nákladov podľa schváleného rozpočtu OI.

Učiteľom informatiky:

Sprístupnite, prosím, tento materiál vašim žiakom. Ideálne by bolo spraviť niekoľko fotokópií zadaní a rozdať ich záujemcom, prípadne vyvesiť kópiu zadaní na nástenku. Podporte účasť študentov v našej súťaži. Môžu tak získať cenné vedomosti siahajúce nad rámec stredoškolských osnov. Možno práve na Vašej škole vyrastá budúci účastník Medzinárodnej informatickej olympiády. Mnohí z nich získavali skúsenosti práve účasťou v našom seminári.

Stredoškólákom:

Ak si žiakom strednej školy a vieš aspoň trochu programovať, neváhaj a zapoj sa do našej súťaže, má to množstvo výhod:

- Môžeš si porovnať svoje schopnosti so žiakmi z celého Slovenska.
- Riešením súťažných úloh a štúdiom poskytnutých vzorových riešení sa môžeš naučiť mnoho nových vecí. Získané poznatky a skúsenosti sa Ti iste budú hodiť na iných súťažiach v programovaní, pri prijímacích pohovoroch i počas vysokoškolského štúdia.
- Na riešenie úloh máš dosť času a môžeš ich riešiť doma bez toho, aby si niekam cestoval.
- Medzi zadaniami sa nachádzajú ľahšie aj ťažšie. Každý si môže vybrať tie, ktoré vie riešiť a ktoré sa mu zdajú zaujímavé.

- Najlepších riešiteľov pozývame každoročne na dve týždenné sústredenia. Sústredenie je jedinečná príležitosť ako spoznať nových priateľov s podobnými záujmami, naučiť sa veľa nového a zažiť veľa zábavy.

Rodičom a sponzorom:

Organizácia korešpondenčného seminára a sústredení pre najlepších riešiteľov je z finančného hľadiska čím ďalej, tým náročnejšia. Preto s radosťou uvítame každého, kto by bol ochotný KSP sponzorovať. V prípade, že by ste KSP chceli sponzorovať a chcete presnejšie informácie, prípadne o nejakom sponzorovi viete, ozvite sa prosím mailom na ksp-info@ksp.sk.

.....

Priezvisko a meno: _____

Dátum narodenia: _____

Rok maturity: _____

Škola: _____

Adresa pre korešpondenciu:

Adresa domov:

E-mail: _____

Telefón: _____

27. ročník KSP – evidenčný lístok

.....

Priezvisko a meno: _____

Dátum narodenia: _____

Rok maturity: _____

Škola: _____

Adresa pre korešpondenciu:

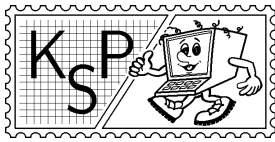
Adresa domov:

E-mail: _____

Telefón: _____

27. ročník KSP – evidenčný lístok

.....



Korešpondenčný seminár z programovania XXVII. ročník, 2009/10

Katedra základov a vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

*KSP finančne podporujú: MICROSTEP-MIS spol. s r.o.
Agentúra na podporu výskumu a vývoja*

Pravidlá

Korešpondenčný seminár z programovania (KSP) je súťaž programátorov – stredoškôľakov (a mladších). Jej cieľom je zdokonaľiť žiakov, ktorí už vedú programovať, alebo sa práve programovať učia, v programovaní a v algoritmickej mysli.

Kategórie: Korešpondenčný seminár z programovania pozostáva z troch kategórií: **KSP-Z**, **KSP-O** a **KSP-T**. Kategória KSP-Z je určená pre začínajúcich riešiteľov. Pokročilým riešiteľom je určená kategória KSP-O. Tým najlepším odporúčame riešiť navyše aj kategóriu KSP-T. **Na sústredenie však na základe riešení úloh z kategórie KSP-T nepozývame – úlohou tejto kategórie je celoročná príprava riešiteľov na medzinárodné súťaže.**

Kategórií KSP-O a KSP-T sa môžu zúčastniť všetci stredoškôľáci (a základnoškôľáci) bez obmedzenia, aj riešitelia KSP-Z. Obzvlášť radi privítame účasť dievčat.

Kategóriu KSP-Z v tomto polroku môžu riešiť:

- druháci na strednej škole a mladší, ak do 1.9.2009 boli najviac na 1 sústredení KSP
- tretiaci a štvrtáci na strednej škole, ak do 1.9.2009 neboli na sústredení KSP

Za štvrtákov považujeme študentov, ktorí maturujú v tomto školskom roku, tretiaci sú tí, ktorí budú maturovať budúci školský rok, ostatné ročníky analogicky.

Odmena: Najlepších riešiteľov z kategórií KSP-Z a KSP-O pozývame každoročne na dve týždenné sústredenia (pokiaľ sa ich podarí zorganizovať). Na jesenné sústredenie nepozývame riešiteľov, ktorí už ukončili strednú školu.

Najlepších riešiteľov z kategórie KSP-T čakajú hodnotné vecné ceny (ako napríklad fajné knihy, štýlové tričká a tak, veď uvidíte).

Pokiaľ viete o niekom, kto by mohol sústredenia KSP čo i len čiastočne sponzorovať, dajte nám o ňom (a prípadne jemu o nás) vedieť.

Organizácia súťaže: Všetky kategórie sú organizované v štyroch kolách, ktoré sú rozdelené na dva samostatné polroky po dvoch kolách. Na jarné sústredenie teda budú pozývaní najlepší riešitelia podľa počtu bodov za zimnú časť a na jesenné sústredenie podľa počtu bodov za letnú časť. Kolo obsahuje 10 príkladov, z čoho **príklady 1-5 tvoria kategóriu KSP-Z, príklady 4-8 kategóriu KSP-O a príklady 8-10 kategóriu KSP-T**. Príklady každého kola (nie nutne všetky) je treba vyriešiť a do určeného termínu poslať na našu adresu. Riešenia odoslané po tomto termíne nemusia byť hodnotené, prípadne môžu byť riešiteľovi strhnuté body za neskoré odoslanie riešenia. Riešiteľom, ktorí sa zúčastnili niektorého kola súťaže, pošleme zadania ďalšieho kola, výsledkovú listinu a komentáre k riešeniam. Ďalšieho kola sa však môžu zúčastniť aj tí žiaci, ktorí sa predchádzajúceho kola nezúčastnili.

Požiadavky na riešenia: Základom riešenia každého príkladu (ak nie je v zadaní uvedené inak) je **listing programu** v ľubovoľnom vyššom programovacom jazyku (najlepšie Pascal, C/C++). Na programe nás zaujíma najmä jeho korektnosť (dáva program pre každý vstup správny výsledok?) a efektivita (koľko času a pamäte potrebuje na spracovanie vstupu určitej veľkosti). Dôraz kladte na algoritmickej časť programu, pri načítavaní vstupu a vypisovaní výstupu nemusíte úplne presne dodržiavať formát zo zadania. Rovnako dôležitou súčasťou riešenia je **slovný popis** riešenia. Slovný popis by mal obsahovať popis použitého algoritmu,

prípadne popis kľúčových dátových štruktúr. Mal by byť natoľko jasný a zrozumiteľný, aby bolo podľa neho možné napísať program rovnako efektívny, ako ten váš. Ďalej vyžadujeme **zdôvodnenie** (dôkaz) **správnosti** použitého algoritmu a odhad časovej a pamäťovej zložitosti algoritmu (t.j. koľko času a pamäte potrebuje váš program v závislosti od veľkosti vstupných dát).

Na internetovej stránke <http://www.ksp.sk/ksp/riesenie.php> nájdete ukážkovo vyriešených niekoľko starších príkladov. Na tej istej stránke sa môžete aj dočítať, čo je vlastne časová a pamäťová zložitosť (ak vám tieto pojmy veľa nehovoria).

Technické detaily: Podobne ako v olympiáde z informatiky, ak píšete program v C++ a používate STL, súčasťou popisu vášho algoritmu by mal byť dostatočne podrobný popis implementácie dátových štruktúr, ktoré používate. Podobné podmienky platia aj pre používanie pokročilých dátových štruktúr v Jave či inom programovacom jazyku. Ak si nie ste istí, či niečo môžete použiť, radšej to popíšte, prípadne nás kontaktujte mailom na ksp-info@ksp.sk a opýtajte sa.

Spôsob doručenia: Diskety nám neposielaťe. Svoje riešenia posielaťe buď na papieri poštou, alebo použite formulár na adrese <http://www.ksp.sk/ksp2.0/eRiesenie>.

Formálna úprava papierových riešení:

Riešenia píšete na papieri formátu A4. Prvý list každého riešenia má obsahovať hlavičku, v ktorej je meno riešiteľa, číslo príkladu, kategória, počet listov príkladu a škola. Ďalšie listy obsahujú meno, číslo príkladu a číslo listu. Ak má príklad niekoľko listov, zopnite ich spolu (v ľavom hornom rohu), aby sa nestratili. Nezopínajte listy patriace rôznym príkladom a už vôbec nepíšete riešenia viacerých príkladov na ten istý list papiera.

Formálna úprava elektronických riešení:

Akceptujeme riešenia len v textovom formáte alebo vo formáte pdf. Ak odovzdávate vo formáte pdf, zdrojový kód odovzdajte aj v textovom formáte. Každý príklad odovzdávajte v samostatnom súbore. Hlavička jednotlivých strán a formát strán je rovnaký ako pri papierových riešeniach.

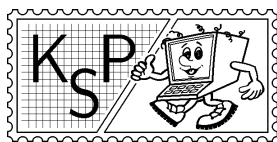
Dôležité: Pred odovzdaním elektronického riešenia je potrebné sa zaregistrovať a vyplniť požadované kontaktné údaje. Registrácie sa potvrdzujú a preto sa zaregistrujte minimálne týždeň pred dňom, kedy chcete odovzdať vaše riešenia. Riešenia sa odovzdávajú na nasledovnej stránke <http://ksp.sk/ksp2.0/eRiesenie>.

Bodovanie: Maximálny počet bodov za príklad závisí od jeho náročnosti, uvádzame ho pri zadaní každého príkladu. KSP je súťaž jednotlivcov, preto ak sa vyskytnú „kolektívne riešenia“, bude za ne udelených patrične menej bodov.

Evidenčný lístok: Každý súťažiaci, ktorý ešte KSP neriešil (alebo sa mu od minula zmenili niektoré údaje) musí poslať spolu s riešeniami prvého kola vyplnený evidenčný lístok. Veľmi dôležitou informáciou na tomto lístku je adresa pre korešpondenciu. Na túto adresu budeme posielať opravené riešenia a zadania ďalších kôl. Ak táto adresa je iná, ako vaša domáca adresa (ak bývate napr. na internáte), napíšte nám aj svoju domácu adresu, aby sme vám mohli písať aj cez prázdniny. Nezabudnite tiež uviesť rok, v ktorom maturujete. Tí z vás, ktorí máte prístup k internetu, môžete evidenčný lístok vyplniť na adrese <http://www.ksp.sk/ksp2.0/ePrihlaska>.

Poštovné náklady: Žiaľ nie je v našich silách hradiť všetky výdavky na poštovné (rozposielanie zadaní, riešení a vzorových riešení). Preto sme zaviedli „účastnícky poplatok“. Účastnícky poplatok je **15 Sk** za každú sériu, ktorej sa riešiteľ zúčastní a posielal sa spolu s **riešeniami úloh formou poštových známok v rozumných hodnotách, napr. 2 a 5 Sk**. Účastníkom, ktorí nám nepošlú známky, hrozí, že od nás nedostanú žiadnu poštu.

Upozornenie: Účasťou v KSP nám zároveň dávate súhlas spracovať a archivovať v elektronickej podobe údaje, ktoré nám poskytnete na evidenčnom lístku, ako aj súhlas zverejniť vaše meno, školu, ročník a dosiahnuté výsledky vo výsledkovej listine.



Korešpondenčný seminár z programovania
XXVII. ročník, 2009/10
Katedra základov a vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

*KSP finančne podporujú: MICROSTEP-MIS spol. s r.o.
Agentúra na podporu výskumu a vývoja*

Príklady 1. kola letnej časti

Milé naše riešiteľky, milí naši riešitelia,
Užite si prvú sériu letnej časti.

Termín odoslania riešení tejto série je pondelok **8. marca 2010.**

Kto sa veľmi namáha, skoro ustane.

KSPáci

1. Zablatená cesta

kat. Z; 10 bodov

Aj keď sa to možno nezdá, Petržlen je cyklistického neba hviezda. Dobré, troška to preháname. Ale Petržlen si veľmi rád zabicykluje. Dnes sa rozhodol, že by sa rád na bicykli vydal do Bratislavy. Aby si nezničil nohy, chce sa počas celej cesty stále namáhať rovnako. No okolo Bratislavy je veľa blata a navyše nerovnomerne rozdeleného. A čím ho je viac, tým ťažšie sa po ňom ide.

Aby sa cestou Petržlen nenudil, zoberie si MP3 prehrávač, v ktorom má svoj obľúbený playlist. Teraz sa ale trápi: ako veľmi má makať, aby mu cesta do Bratislavy trvala presne toľko isto ako je dĺžka jeho playlistu? Navyše v najnovších časopisoch sa dočítal, že má makať rovnomerne, lebo vtedy sa najmenej unaví.

Úloha: Petržlenov playlist má dĺžku T hodín. Cesta do Bratislavy je tvorená N rovnako dlhými úsekmi. Každý z úsekov má dĺžku D . Na úseku číslo i je množstvo blata rovné k_i . Ak bude Petržlen šľapať tak, že po úseku bez blata pôjde rýchlosťou v , tak po úseku kde je k_i blata pôjde rýchlosťou $v \left(1 - \frac{k_i}{4+k_i}\right)$.

Napište program, ktorý načíta čísla T , N , D a k_1, \dots, k_N a vypočíta takú rýchlosť v , pre ktorú bude Petržlenovi celá cesta trvať presne T hodín. Odpoveď stačí vypísať vhodne zaokrúhlenú.

Príklad:

Vstup:

3 2 5
0 4

Výstup:

5.000000

(Prvým úsekom pôjde rýchlosťou 5 km/h, druhým rýchlosťou 2.5 km/h. Prvý úsek prejde za hodinu a druhý za dve.)

Vstup:

7 4 12
4 8 0 11

Výstup:

16.714286

(Pre $v \simeq 16.7$ budú Petržlenovi jednotlivé úseky trvať približne 1.44, 2.15, 0.72 a 2.69 hodiny.)

2. Zmrzlináreň

kat. Z; 10 bodov

Nedávno sa na matfýze objavila zmrzlináreň. Má veľký úspech, lebo tam predávajú zmrzlinu s rôznymi exotickými príchutami a dokonca aj s rôznymi tvarmi kopčeka. Okrem tradičných guľových kopčiek sú v ponuke kockové, valcové, ihlanové, ba dokonca aj trojstenové!

Najdrsnejšia príchuť v ponuke je karamelovo-muchotrávková, druhá je slivkovo-rybacia a tretia najdrsnejšia je punčovo-paštétová. Tieto tri príchute dokážu občas poriadne zatočiť žalúdkom. Ostatné príchute nie sú vôbec drsné a žalúdkom netočia.

Presnejšie, funguje to nasledovne. Vždy, keď príde do žalúdka kopček zmrzliny, pozrie sa na všetky kopčeky, ktoré tam už sedia. A vždy, keď uvidí kopček, ktorý má menej drsnú príchuť ako on sám, tak raz zatočí žalúdkom.

Úloha: Maják práve nalačno zjedol N zmrzlín. Na vstupe je číslo N a postupnosť príchuť zmrzlín v poradí, v akom ich Maják zjedol. Napíšte program, ktorý spočíta, koľkokrát sa Majákovi počas jedenia zatočil žalúdkom.

Príklad:

Vstup:

7

karamelovo-muchotravkova

slivkovo-rybacia

puncovo-pastetova

slivkovo-rybacia

jahodova

malinova

karamelovo-muchotravkova

Výstup:

6

(Prvé tri kopčeky Majákovi žalúdok nezatočili, lebo každá príchuť je menej drsná ako tie pred ňou. Štvrtý kopček mu zatočil žalúdok $1\times$. Piaty a šiesty kopček sú obyčajné príchute, ktoré žalúdkom netočia. No a posledný, siedmy kopček zatočí žalúdkom hneď $5\times$.)

3. Zvukové bariéry

kat. Z; 10 bodov

V Absurdistane, krásnej to krajine, vedie diaľničný obchvat hlavného mesta absurdne krížom cez mesto a význam slova „obchvat“ tým dostal jedinečne národný charakter. Obyvateľom z tesnej blízkosti obchvatu sa však vôbec nepáčil hluk áut, ktorý je slýchať vo dne i v noci.

Pod tlakom obyvateľstva sa tak Národná Diaľničná Spoločnosť Absurdistanu rozhodla postaviť okolo obchvatu zvukové bariéry. Aby sa podporila domáca výroba, nakúpili izolačné bloky od domácej firmy a tak sú tieto bloky rozličných i dosť divokých tvarov. Z toho inžinieri veru radosť nemali; veď uznajte, ako z takých exotických blokov postaviť pekný múr? Inžinierom sa to riešiť veru nechcelo, nuž rozhodli sa nechať to na žeriavnikov, veď oni to už dáko poskladajú. Chyba. Žeriavnici, nemajúc nad sebou kontrolu, ukladali dielce tak, ako im pod ruku, či vlastne pod hák¹ prišli.

Stimulovaní týmito udalosťami sa inžinieri rozhodli žeriavnikov kontrolovať, aby múry konečne boli múrmi. Robiť to osobne sa im však samozrejme nechce, ale už to nemôžu nechať na náhodu a tak sa rozhodli zveriť túto úlohu niekomu z radov zahraničných odborníkov. Spomedzi všetkých si vybrali práve teba! Veríme, že nám v Absurdistane neurobiš hanbu.

Úloha: Teraz trocha formálnejšie. Na prvom riadku vstupu sú 4 prirodzené čísla: dĺžka múru D , výška múru V , počet typov blokov T a počet blokov N , ktoré žeriavnik na stavbu múru použil.

Nasleduje popis T typov blokov v poradí $1 \dots T$: šírka A_i a výška B_i i -teho typu a B_i riadkov, každý po A_i znakoch „X“ alebo „-“ popisujúcich tvar bloku. Môžete predpokladať, že každý blok je súvislý², neprázdny a nie je väčší ako samotný múr.

Nasleduje N riadkov: postupnosť, ako žeriavnik ukladal bloky. Riadok, ktorý je j -ty v poradí obsahuje dve prirodzené čísla, a to polohu P_j , kam bol umiestnený blok typu R_j ($1 \leq P_j \leq D$ a $1 \leq R_j \leq T$). Ak hovoríme, že blok bol položený na polohu P_j , myslíme tým, že na túto pozíciu bol zarovnaný ľavý bok tohto bloku a ten bol takto položený. Môžete tiež predpokladať, že žiaden blok neprekročí dĺžku múru, výšku ale môže.

¹Hádanka: Ruka, klepeto i hák a pritom to nie je pirát! Kto je to?

²Ak je blok väčší ako 1×1 , každé „X“ sa dotýka aspoň jedného iného „X“ hranou, nie len rohom.

Vašou úlohou je odpovedať „ANO“ alebo „NIE“ na otázku, či sa zadanou postupnosťou postavil súvislý múr – obdĺžnik, ktorý nemá diery a má rozmery presne $D \times V$.

Príklad:

Vstup:

$D = 4 \ V = 2 \ T = 2 \ N = 2$

4 1

XXXX

1 1

X

1 2

1 1

Vstup:

$D = 3 \ V = 2 \ T = 2 \ N = 3$

3 2

X--

XXX

1 1

X

1 1

2 2

3 2

Výstup:

NIE

chceme

XXXX

XXXX

dostaneme ale

XXXX

X---

Výstup:

ANO

..... Tu sa začínajú zadania KSP-O

4. Zaplať a leť

kat. Z a O; 15 bodov

Arina je veľmi sporivá žienka, avšak keď zbadala reklamu na novú leteckú spoločnosť LAMA³, nemohla odolať. Ponúkajú totiž lety na stratené letiská pomocou magických lietadiel. V jednom okamihu ste na jednom letisku a v tom druhom už ste na inom letisku, ani sa nenazdáte.

No odolali by ste? Arina totiž vždy chcela vidieť letisko spoločnosti IBM⁴. A ono je dokonca v ponuke! Avšak ako každá novinka, ani tieto lety nie sú práve najlacnejšie a dokonca nie vždy existuje priamy let. Preto by potrebovala vedieť, koľko ju to bude najmenej stáť, aby mohla začať šetriť.

Preto si opísala všetky lety, čo existujú, a zistila, že letisko IBM je tzv. koncové letisko, čiže je to letisko, z ktorého sa už letieť inam touto spoločnosťou nedá. Dokonca je jediné také. Ó, tá pocta! Ale čože je toto, všetky letiská majú dokonca svoje číslo a lety vedú len na niektoré z ďalších letísk, nikdy nie späť. To len potvrdzuje staré príslovie „Všetky cesty vedú do IBM!“ Arina sa pousmiala, ale potom zazúfala; kto jej pomôže a povie, koľko si má našetriť? Preto sa obrátila na vás, milí KSPáci. . .

Úloha: Na vstupe dostanete N (počet letísk) a K (počet liniek, ktoré medzi nimi premávajú). Na ďalších K riadkoch budú popisy letov v tvare „ $i \ j \ c$ “ (bez úvodzoviek), ktoré sú utriedené vzostupne podľa i (a následne aj podľa j); $i < j$ a let vedie z letiska i na letisko j a stojí c .

Na výstup vypíšte, koľko stojí najlacnejšia cesta z letiska 1 na letisko N . Ak taká neexistuje, vypíšte string „neexistuje“.

Môžete predpokladať, že $2 \leq N \leq 10^6$, $0 \leq K \leq 2 \cdot 10^6$, $0 \leq c_i \leq 1000$ a že medzi každou dvojicou staníc existuje maximálne jeden let.

³Lost Airports Magic Airlines

⁴Institute of Black Magic

Odobzďavanie: Tento príklad je praktický, čo znamená, že sa dá odobzďavať jedine elektro-
nicky, pričom stačí odobzďať iba zdrojový kód. Pri odobzďaní sa vám program otestuje na
niekoľkých vstupoch a dostane body podľa toho, na koľkých vstupoch dá správny výsledok
v časovom limite.

Príklad:

Vstup:
4 6
1 2 4
1 3 3
1 4 15
2 3 1
2 4 10
3 4 2

Výstup:
5

5. Opravovacia lotéria vracia úder

kat. Z a O; 15 bodov

V prvej sérii sme boli svedkami toho, ako sa lotéria opravovateľského Impéria zrútila ako
domček z karát. Impérium sa ale rozhodlo, že to len tak nenechá a pripravilo novú, lepšiu
lotériu. Ste pripravení zasadiť jej úder priamo do jadra?

V tejto úlohe sa opäť rozďávajú body zadarmo. Stačí, keď nám pošlete reťazec 32 zna-
kov malej anglickej abecedy (napr. „abcdefghijklmnopqrstuvwxyza“) a môžete aj vy
nejaké body vyhrať!

Počet bodov, ktoré dostanete, bude rovný tomu, čo pre vaše číslo vypíše nasledujúci
program:

Listing programu:

```
#include <stdio>
#include <math>

int sbox1[] = {4, 10, 11, 15, 24, 16, 17, 1, 6, 9, 3, 22, 19, 2, 0, 23, 25,
18, 13, 8, 5, 21, 7, 12, 14, 20};
int sbox2[] = {2, 0, 20, 13, 11, 17, 4, 3, 9, 18, 23, 15, 24, 6, 16, 14, 22,
10, 25, 12, 21, 8, 7, 5, 19, 1};

#define NUM_ROUNDS 223456789012311

char key[] = "bazavektorovehopriestoru";

long long countpoints(char *a)
{
    long long ret = 0;
    for(int i = 0; i < 32; i++)
        for(int j = i + 1; j < 32; j++)
            if(sbox1[a[i]-'a'] == sbox2[a[j]-'a'])
                ret++;
    return ret;
}

void round(char *in, char *out)
{
    for(int i = 0; i < 16; i++)
        out[i+16] = 'a' + sbox2[in[i]-'a'];
    for(int i = 16; i < 32; i++)
        out[i-16] = 'a' + sbox1[(in[i-16]-'a' + in[i]-'a' + key[i-16]-'a') % 26];
    out[32] = 0;
}
```



```

}

int main()
{
    char text[2][35];
    int ind = 0;
    scanf("%s", text[ind]);
    for(long long i = 0; i < NUM_ROUNDS; i++){
        round(text[ind], text[1 - ind]);
        ind = 1 - ind;
    }
    printf("%s\n", text[ind]);
    printf("%d\n", (int)sqrt(countpoints(text[ind]) / 2.0 + 8) - 1);
}

```

Program je písaný v C++. Ak budete dobre hľadať, na webstránke KSP nájdete aj ekvivalentné programy napísané v Pascale a možno aj v iných jazykoch⁵.

V tejto úlohe nám neposielajte žiadne programy. Jediné, čo budeme hodnotiť, je konkrétny reťazec, ktorý odovzdáte. (Neurazíme sa, ak nám k nemu napíšete pár viet o tom, ako ste sa k tomu konkrétnemu reťazcu dostali.)

..... Tu sa končia zadania KSP-Z

6. O kráľových jedoch

kat. O; 20 bodov

Kráľ Baltazár (prezývaný tiež Baltýk) usporiadal obrovskú kráľovskú hostinu, na ktorú pozval všetkých svojich priateľov aj nepriateľov, skrátka každého. A ako správny hosťiteľ, chcel na nej otráviť všetkých svojich nepriateľov. Plán bol veľmi jednoduchý. Každý hosť donesie na hostinu fľašu vína. Priatelia určite donesú kvalitné víno, nepriatelia zase určite donesú kvalitne otrávené víno. Stačí teda každému dať sa napiť vína, ktoré priniesol.

Hostina začala, ale čo sa nestalo. Kvalitné vína sa pomiešali s kvalitne otrávenými tak, že nik nevedel povedať, ktoré je ktoré. Našťastie má kráľ niekoľko sluhov, ktorým môže dať ochutnať víno, prípadne zmes vín. Sluha po ochutnaní zomrie práve vtedy, ak aspoň jedno z vín v zmesi, ktorú pil, bolo otrávené. Kráľovi je testovanie vína na sluhoch nepríjemné, preto by chcel otrávené vína nájsť na čo najmenej pokusov.⁶

Úloha: Máme N vín, z čoho práve K je otrávených. Vašou úlohou je napísať program, ktorý bude radiť kráľovi, ako má testovať vína tak, aby identifikoval všetky otrávené vína. Vína sú pre jednoduchosť očíslované od 1 po N . Program bude musieť fungovať nasledovne:

1. Načíta N a K .
2. Vypíše počet M a čísla v_1, v_2, \dots, v_M vín, ktoré sa majú zmiešať a dať vypiť sluhovi.
3. Načíta, či sluha po vypití zmesi vín zomrel.
4. Ak vie, ktoré vína sú otrávené, tak vypíše ich čísla a skončí, inak pokračuje krokom 2.

Bodovanie: Prvoradým kritériom hodnotenia bude počet pokusov, druhoradým časová a pamäťová zložitosť. Pri hodnotení počtu otázok ignorujeme konštanty, podobne ako pri časovej a pamätevej zložitosti. Riešenie potrebujúce 47-krát viac otázok ako optimálne teda môže získať plný počet bodov, ale riešenie s K -krát viac otázkami už nie.⁷ Navyše predpokladáme, že K je ďaleko menšie ako N .

⁵Brainfuck ale nečakajte

⁶počet pokusov nie je to isté ako počet mŕtvych sluhov

⁷Formálne: Nech $f(N, K)$ je počet pokusov, ktoré potrebuje algoritmus A pri N vínach, z ktorých je K otrávených. Potom každý algoritmus, ktorý potrebuje $\Theta(f(N, K))$ pokusov, budeme považovať za rovnako dobrý, čo sa počtu otázok týka.

Súčasťou riešenia by mal byť aj odhad počtu pokusov v závislosti od N a K , aj s dôkazom, že je odhad správny. Samozrejme, program musí byť správny a teda vždy označiť ako otrávené práve tie vína, ktoré sú otrávené.

Príklad:

program: 3 1 2 3	vstup: 10 2
program: 3 1 4 5	vstup: sluha zomrel
program: 4 1 3 4 7	vstup: sluha zomrel
program: otravene su 2 a 5	vstup: sluha nezomrel

7. O godzillích dostihoch

kat. O; 20 bodov

Mišo s Majákom si zasa vymysleli novú hru. Jej pravidlá sú veľmi jednoduché:

Hra je určená pre 2 hráčov. Hrací plán sa skladá z radu $N + 1$ políčok, ktoré sú očíslované zaradom od 0 po N . Na políčko číslo N sa položí figúrka Godzilly. Táto figúrka má na začiatku hry rýchlosť 0. Hráči teraz striedavo ťahajú. Ťah hráča spočíva v tom, že sa rozhodne, či chce rýchlosť figúrky o 1 zvýšiť, o 1 znížiť, alebo nechať nezmenenú. (Pritom ale počas hry musí byť rýchlosť stále kladná.) Godzillu hráč následne pohne smerom k políčku 0 toľkokrát, aká je jej aktuálna rýchlosť.

Hru vyhrá ten, kto vo svojom ťahu dosiahne políčko 0 (prípadne prejde cez neho), a to aj vtedy, keby ešte mal v danom ťahu pohyb Godzilly pokračovať.

Príklad: Godzilla stojí na políčku 11 a jej aktuálna rýchlosť je 7. Na ťahu je Mišo. Rozhodne sa znížiť rýchlosť na 6 a následne posunie Godzillu na políčko číslo $11 - 6 = 5$. V nasledujúcom ťahu Maják vyhrá, bez ohľadu na to, ako sa rozhodne zmeniť rýchlosť.

Hra vzbudila značný dojem a ostatní KSPáci začali tipovať, kto kedy vyhrá. Mio sa rozhodol na tejto hre zbohatnúť, no tipovanie mu veľmi nejde, preto vás žiada o radu.

Úloha: Vašou úlohou je napísať program, ktorý pre dané N vypíše, ktorý hráč vyhrá, ak budú hrať obaja optimálne. Hru vždy začína Mišo.

Príklad:

Vstup:
8

Výstup:
Majak vyhra.

..... Tu sa začínajú zadania KSP-T

8. Tragédia

kat. O a T; 25 bodov

Ach, kam sa to zase dostal? Georg⁸, svetoznámy scenárista a režisér, sa po riadnom kopnutí múzou ocitol uprostred ničoho, v obci Nalomená Trieska. Myslel si, že odrezaný od civilizácie sa bude môcť plne sústrediť na svoju najnovšiu autobiografickú jednoaktovku s pracovným názvom „Utrpenie mladého Georga“. To sa mu síce podarilo a dokonca už má náčrt scenára, ale zabudol na jeden neprijemný detail. V Nalomenej Trieske nie je divadlo. Starosta našťastie podporuje umenie a preto poskytol Georgovi na skúšanie šatňu miestnej materskej školy. Tiež sa prihlásilo zopár obyvateľov so záujmom o dosky znamenajúce svet, ktorých budeme pre jednoduchosť nazývať v ďalšom texte hercami (aj keď tak vôbec nevyzerajú).

Začalo sa skúšať. Jednoaktovka sa skladá z jedného dejstva, ktoré sa delí na scény a tie sa delia na vety. Na začiatku každej scény prídu z jedálne (improvizované zákulisie) do šatne (improvizované javisko) herci, ktorí majú počas tejto scény vystupovať. Odohrajú si svoje úlohy, vyrieknu svoje repliky a na konci scény sa znova vrátia do jedálne. No prišla zima a v šatni je zrazu pomenej miesta. Herci naďalej skúšajú, hlava-nehlava, noha-nenoha. Ako

⁸čítaj [georg]

sa tak v zápale dialógov prechádzajú po šatni a prekračujú porozhadzované topánky, občas niektorý herec stúpi inému na nohu. Presnejšie, počas jednej scény stúpi každý herec každému hercovi na nohu práve raz. Aj sebe (naozaj je tam málo miesta). Herci s pošliapanými nohami hrajú o poznanie horšie, preto chce Georg zakročiť. Plánuje rozdeliť svoju jednoaktovku na scény tak, aby bolo dokopy šliapnutí na nohu čo najmenej.

Úloha: Stručne a (snáď) jasne: rozdeľte postupnosť čísel na súvislé úseky tak, aby po sčítaní druhých mocnín počtu rozdielnych čísel v každom úseku vyšiel čo najmenší výsledok.

Podrobnejšie: na vstupe dostanete v prvom riadku dve čísla N (počet viet v Georgovom scenári, $1 \leq N \leq 100000$) a M (počet hercov, $1 \leq M \leq N$) oddelené medzerou. V druhom riadku nasledujú čísla h_i pre $1 \leq i \leq N$ oddelené medzerami; vetu i má vyriešiť herec číslo h_i ($1 \leq h_i \leq M$). Vašou úlohou je nájsť také rozdelenie viet na scény, ktoré minimalizuje celkový počet šliapnutí na nohu a vypísať tento počet. Každá scéna musí pokrývať súvislý úsek viet a každá veta sa musí nachádzať práve v jednej scéne. Počas scény, v ktorej vystupuje k hercov, nastane k^2 šliapnutí na nohu. A samozrejme, kto nemá v scéne vetu, nevystupuje v nej (teta kuchárka občas nechá v jedálni chladnúť pečené buchty a vtedy je problém dostať na javisko aj tých, ktorí majú v scéne vystupovať).

Odvzdávanie: Tento príklad je praktický, čo znamená, že sa dá odovzdávať jedine elektronicky, pričom stačí odovzdať iba zdrojový kód. Pri odovzdaní sa vám program otestuje na niekoľkých vstupoch a dostane body podľa toho, na koľkých vstupoch dá správny výsledok v časovom limite.

Príklad:

Vstup:

7 4
3 1 4 1 1 4 1

Výstup:

5
Optimálne je rozdelenie na dve scény. V prvej bude len prvá veta, ktorú má vyriešiť herec číslo 3 (1 šliapnutie), v druhej scéne budú zvyšné vetty, ktoré majú vyriešiť herci 1 a 4 (4 šliapnutia).

..... Tu sa končia zadania KSP-O

9. Tajný plán škorca Jonáša

kat. T; 25 bodov

Na starej čerešni sa premnožili húsenice. Ale pst, nepovedzte to nikomu! Jonáš je zatiaľ jediný škorec, ktorý o tom vie. A plánuje tam hneď zajtra zaletieť a napráškať sa, čo to dá. Samozrejme si to ale najskôr potrebuje dobre naplánovať. A tu by sa mu zišla vaša pomoc.

Čerešňa je strom. Keď sa na ňu pozeráme ako na graf, má $N - 1$ hrán (konárov a kusov kmeňa medzi vetveniami) a N vrcholov s číslami od 1 po N (miesta vetvenia, konce konárov a miesto, kde sa dotýka zeme). O každej hrane vie Jonáš počet húseníc, ktoré na nej žijú.

Jonášov plán vyzerá nasledovne: pristane na strome vo vrchole A a následne prelezie po hranách do vrcholu B a cestou vyžerie všetky húsenice. Samozrejme, pôjde tak, aby nešiel po žiadnej hrane viackrát (ísť po konári, kde už nie sú húsenice, je strata času).

Úloha: Napíšte program, ktorý načíta popis stromu, spracuje si ho a následne bude Jonášovi odpovedať na otázky. Každá otázka má tvar: „koľko húseníc zožeriem, ak prejdem z vrcholu x_i do vrcholu y_i ?“

Hodnotenie: aby ste dostali nejaké zaujímavé body, je **nutné** splniť nasledujúce dve podmienky: 1. Čas potrebný na spracovanie stromu musí byť lepší ako kvadratický od N . 2. Najhorší čas potrebný na zodpovedanie otázky musí byť lepší ako lineárny od N .

Pri dodržaní týchto dvoch podmienok bude primárnym kritériom hodnotenia časová zložitosť odpovedania na otázky, sekundárnym časová zložitosť predspracovania a terciárnym pamäťová zložitosť.

Príklad:**Vstup:** $N = 7$

hrany:

1 – 4 (10), 2 – 4 (11), 3 – 5 (24),
 4 – 5 (13), 5 – 6 (9), 5 – 7 (10).

otázky:

1 3

2 3

Výstup:

47

48

V popise vstupu „ $x-y(z)$ “ predstavuje hranu medzi vrcholmi x a y so z húsenicami. Formát vstupu si samozrejme zvolte vlastný.

Prvý výstup: Na ceste z 1 do 3 pôjde cez vrcholy 4 a 5 a zožerie $10+13+24=47$ húseníc.

10. Túžba po majetku

kat. T; 25 bodov

Starý a bohatý gróf deMaroschko umrel a zanechal na tomto svete N synov a veľký majetok vo forme obrovského pozemku a nehnuteľností. Nehnuteľností je M a sú porozmiestňované po celom pozemku. Pôda ako taká je v tejto lokalite bezcenná, avšak každá z nehnuteľností predstavuje veľmi atraktívny a hodnotný objekt. Je medzi nimi napríklad ZOO s Godzillami alebo továreň na konzervy s ananásom.

Problém spočíva v tom, že poslednú vôľu nešťastnou náhodou zjedla lama pred tým, ako si ju stačili právnici prečítať. No a ako to už býva, N nenásytných synov chce túto situáciu využiť a uchmatnúť si pre seba čo najviac z majetku starého grófa.

Každý zo synov zaslal písomný nárok, v ktorom sa dožaduje obdĺžnikovej časti pozemku so všetkými nehnuteľnosťami, ktoré sa nachádzajú vnútri alebo na okraji daného územia. Všetci tvrdia, že gróf im to ústne sľúbil a že sa to určite nachádzalo v poslednej vôli. Hlavný problém je, že sa dané nároky prekrývajú. Územia by sa ešte pahlní potomkovia dokázali vzdať, avšak nehnuteľností sa nevzdajú nikdy.

Zamestnanci pozemkového úradu teraz spolu s právnikmi grófových ratolestí sedia nad písomnými nárokmi a snažia sa nájsť riešenie, s ktorým budú všetky zúčastnené strany súhlasiť. Aby mali ľahšiu prácu, potrebovali by o každej nehnuteľnosti vedieť, súčasťou koľkých nárokov je. Detí a nehnuteľností je ale príliš veľa, a preto by potrebovali program. A ten program musí byť rýchly. Pretože keď nebude rýchly, štát utratí veľa za súdne príehady.

Úloha:

Na vstupe sú čísla N a M – počet nárokov a počet nehnuteľností. Nasleduje N riadkov popisujúcich nároky. Každý nárok je popísaný ľavým horným bodom a pravým dolným bodom. Potom nasleduje M riadkov s pozíciami nehnuteľností (nehnuteľnosti si predstavíme ako body v rovine). Pre každú nehnuteľnosť vypíšte, do koľkých nárokov spadá. Všetky súradnice sú celé čísla, ktoré v absolútnej hodnote neprevyšujú 1,000,000,000. Pri vymýšľaní algoritmu pamätajte na to, že N aj M môže byť až do 100,000. Áno, deMaroschko bol naozaj výnimočný človek. . .

Príklad:**Vstup:** $N = 3 \quad M = 5$

1 3 4 1

3 5 7 2

4 6 5 3

1 4

2 3

4 3

5 2

5 4

Výstup:

0 1 3 1 2