

Korešpondenčný seminár z programovania XIX. ročník, 2001/2002

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

*KSP finančne podporuje nadácia Open Society Fund Bratislava
IUVENTA – zariadenie pre voľný čas detí, mládeže a dospelých
MICROSTEP spol. s.r.o.*

Propozície

Korešpondenčný seminár z programovania (KSP) je súťaž programátorov – stredoškóľakov. Jej cieľom je zdokonaľiť žiakov, ktorí už vedia programovať alebo sa práve programovať učia, v programovaní a v algoritmickom myslení.

Kategórie: Korešpondenčný seminár z programovania pozostáva z dvoch kategórií – **KSP a KSP-Z**. Kategória KSP-Z je určená pre začínajúcich riešiteľov. V tomto ročníku sa jej môže zúčastniť každý stredoškóľák, ktorý v žiadnom z predchádzajúcich ročníkov v kategórii KSP nezískal za celý ročník viac ako **30 bodov**, v kategórii KSP-Z viac ako **90 bodov**, a nezúčastnil sa sústredenia KSP. Kategórie KSP sa môžu zúčastniť všetci stredoškóľáci bez obmedzenia (aj riešitelia KSP-Z). Obzvlášť radi privítame účasť dievčat.

Sústredenia: Najlepších riešiteľov oboch kategórií pozývame každoročne na dve týždenné sústredenia (pokiaľ sa ich podarí zorganizovať). Na jesenné sústredenie nepozývame riešiteľov, ktorí už ukončili strednú školu. Pokiaľ viete o niekom, kto by mohol tieto sústredenia čo i len čiastočne sponzorovať, dajte nám o ňom (a prípadne jemu o nás) vedieť.

Organizácia súťaže: Kategória KSP aj kategória KSP-Z 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 1. a 2. kolo zimnej časti a na jesenné sústredenie podľa počtu bodov za 1. a 2. kolo letnej časti. Každé kolo obsahuje päť úloh. 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 algoritmickú časť programu, pri načítavaní vstupu a vypisovaní výstup 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 pamätevej zložitosti algoritmu (t.j. koľko času a pamäte potrebuje váš program v závislosti od veľkosti vstupných dát).

Formálna úprava: **Diskety nám neposielaťe.** Svoje 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 nepíšte riešenia viacerých príkladov na ten istý list papiera.

Bodovanie: Maximálne počty bodov pre jednotlivé príklady sa v zadaniach neuvádzajú, pohybujú sa od 10 do 25 bodov podľa náročnosti príkladu. KSP je súťaž jednotlivcov, preto ak sa vyskytnú „kolektívne riešenia“, bude za ne udelených patrične menej bodov.

Evidenčný lístok: Každý súťažiaci musí poslať spolu s riešeniami prvého kola vyplnený evidenčný lístok. Veľmi dôležitou informáciou na tomto lístku je adresa pre korešpondenciu. Na túto adresu budeme posielat 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 ročník štúdia a označiť kategórie, ktorých sa chcete zúčastniť. Tí z vás, ktorí máte prístup k Internetu, môžete evidenčný lístok vyplniť na adrese <http://www.ksp.sk/ksp/prihlaska.php>.

Poštovné náklady: Žiaľ nie je v našich silách hradiť všetky výdavky na poštovné (rozposielanie zadání, riešení a vzorových riešení). Preto sme zaviedli „účastnícky poplatok“. Účastnícky poplatok je **15,- Sk** za každú sériu, ktorej sa riešiteľ zúčastní a posiela 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.

Tešíme sa na vaše riešenia a prajeme vám veľa dobrých nápadov

KSPáci

Adresa: KSP

Katedra vyučovania informatiky FMFI UK

Mlynská dolina

842 48 Bratislava 4

E-mail: ksp@ksp.sk

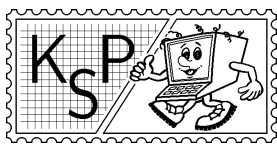
URL: <http://www.ksp.sk/ksp>

Telefón: 02 / 60 29 52 10

Mailing list eKSPres

Mailing list eKSPres je určený riešiteľom a opravovateľom KSP, ako aj najrôznejším priaznivcom seminára. Dá sa na ňom dozvedieť o rôznych akciách organizovaných vedúcimi KSP (napríklad splav), diskutovať o zaujímavých problémoch (s výnimkou príkladov aktuálnej série KSP!) a pod.

- Ak sa chcete doňho prihlásiť, musíte mať e-mailovú adresu. Z nej pošlite na adresu **eKSPres-request@ksp.sk** mail obsahujúci jediné slovo **SUBSCRIBE** v tele správy. Od tohoto okamihu vám budú na vašu adresu chodiť všetky správy z mailing listu eKSPres.
- Ak chcete ostatným poslať mail, pošlite mail na adresu **eKSPres@ksp.sk**. Váš mail príde všetkým členom eKSPres-u, preto sa pred jeho odoslaním najprv zamyslite nad jeho vhodnosťou.



Korešpondenčný seminár z programovania
XIX. ročník, 2001/2002
Katedra vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

KSP finančne podporuje nadácia Open Society Fund Bratislava
IUVENTA – zariadenie pre voľný čas detí, mládeže a dospelých
MICROSTEP spol. s r.o.

Príklady 1. kola letnej časti

Milé naše riešiteľky, milí naši riešitelia,
dostávajú sa vám do rúk zadania 1. kola letnej časti 19. ročníka KSP. Prečítajte si pozorne propozície súťaže a hor sa riešiť. V prípade, že riešite KSP po prvý raz, odporúčame vám začať s kategóriou KSP-Z. Svoje riešenia nám posielajte najneskôr do **11. marca**. Nezabudnite k nim priložiť známky v hodnote 15,- Sk.

Harry Potter sa nesmie dotknúť páchnuceho slimáka!

KSPáci

1. O Kleofášových pozemkoch

Výskumník Kleofáš konečne dosiahol veľký úspech. Na kiribatskom ostrove Dlhý Had objavil niekoľko nálezísk vzácnych fosílií trilobytov (čítaj trilobajtov). Aby ho o jeho majetok nik nepripravil, rozhodol sa, že pozemky, na ktorých náleziská ležia, kúpi. Nie je to však také jednoduché – narazil totiž na odpor kiribatských byrokratov.

Ostrov Dlhý Had má tvar obdĺžnika, ktorého šírka je zanedbateľne malá. Polohu náleziska preto môžeme reprezentovať jediným číslom, ktoré určuje jeho vzdialenosť od začiatku ostrova. Podobne súvislý pozemok na ostrove môžeme reprezentovať uzavretým intervalom. Cena pozemku je priamo úmerná jeho dĺžke. Aby kiribatskí byrokrati nemali priveľa práce, vydali nariadenie, že jeden človek smie vlastniť najviac k súvislých pozemkov. Kleofáš teda rozmýšľa, akých najviac k súvislých pozemkov má kúpiť, aby vlastnil všetky náleziská a pritom zaplatil čo najmenšiu sumu.

Úloha: Napíšte program, ktorý načíta číslo N – počet nálezísk a k – koľko súvislých pozemkov môže Kleofáš kúpiť. Ďalej načíta N čísel, ktoré určujú polohy nálezísk. Program má poradiť Kleofášovi, ktoré pozemky má kúpiť. To znamená, že má vypísať najviac k súvislých intervalov takých, aby každé nálezisko ležalo v nejakom z nich a pritom aby bol súčet ich dĺžok čo najmenší.

Príklad:

Vstup:

$N = 3, k = 2$

Náleziská: 1, 2, 10

Výstup:

$\langle 1, 2 \rangle$

$\langle 10, 10 \rangle$

2. O bájnóm poklade

„To je nuda.“, prehlásil Indiana Jones, následne sa uhol dvom guľkám a naďalej utekal uličkami pred skupinou štyroch zabijakov. Za najbližším rohom zobral plechový kryt a bežiac odrazil ďalšie štyri výstrely, ktoré mu smerovali na hrudník. Zo zúfalstva odhodil plech¹, ten po trojitej rotácii udel prvého prenasledovateľa do tváre, zrazil ho na zem a o neho sa

¹bežať s plechom nie je nič moc

potkol aj kumpán bežiaci za ním. Ďalších dvoch banditov sa Indi Junior tiež poľahky zbavil a onedlho už sedel doma v kresle.

Myslíte si, že to len tak, ale mýlite sa. Jeho otec opäť podnikal bádanie po stratenom poklade. Všetky získané informácie si zapísal do denníka, ktorý kvôli obavám zveril svojmu synovi. Junior si z neho po večeroch čítal, a zaujala ho najmä nasledujúca veta: „*Chodba vedúca k pokladu je uzavretá mechanizmom, ktorý dvere otvorí len vtedy, ak na stôl stojaci v ich blízkosti osoba múdra a šľachetná, položí farebné kamene v správnom poradí.*“ Kameňov je tam vraj hojne, jediný problém je vybrať správne farby a potom ich správne zoradiť. Starý Jones samozrejme zistil, aká postupnosť kameňov otvára dvere, ale z obáv o poklad ju radšej do denníka popísal iba „približne“, pre istotu však dvakrát. Juniора hneď napadlo, že by otec mohol celú postupnosť zabudnúť. Keby popisom v denníku vyhovovala iba jediná postupnosť kameňov, všetko by bolo v pohode, ale čo keď je takých viac? Preto sa obrátil na vás, aby ste mu s týmto problémom pomohli.

Úloha: Je dané číslo N udávajúce počet farieb kameňov. Farby kameňov sú označené číslami $0 \dots N - 1$. Ďalej sú dané dva popisy tej istej postupnosti kameňov skladajúce sa z čísel a malých písmen. Písmená predstavujú jednotlivé úseky, pričom všetky úseky označené jedným písmenom sú rovnako dlhé a popisujú takú istú postupnosť kameňov. Číslo v popise hovorí, že na tom mieste je kameň s farbou označenou príslušným číslom². Nakoniec sú na vstupe dané dĺžky jednotlivých úsekov. Váš program má vypísať, koľko rôznych postupností kameňov vyhovuje zároveň obidvom popisom. Ak obidvom popisom zároveň nevyhovuje ani jedna postupnosť, program o tom vypíše príslušnú hlášku.

Príklad:

Vstup:

$N = 2$

$1, a, b, c, 1$

$b, 0, c, c$

$|a| = 2$

$|b| = 1$

$|c| = 3$

Výstup:

Spolu je 1 možnosť rozostavenia kameňov.

(Pozn.: Obidvom popisom v zadaní vyhovuje jedine postupnosť 10111111. Jednotlivé úseky budú vyzeráť nasledovne: $a = 01$, $b = 1$, $c = 111$.)

3. O vláčikovej súprave

Podobne, ako aj ostatné deti, aj Miško dostal na Vianoce veľa darčiekov. Ale žiaden z nich nebol taký krásny, ako elektrický vláčik, ktorý dostal pred deviatimi rokmi. Vtedy si hneď upratal svoju detskú izbu³, aby mohol na dlážke postaviť okružnú železničnú trasu a na nej obrovskú stanicu. Stanica bola ozajstným skvostom celej trasy. Síce do nej vlak mohol prísť iba po jedinej koľaji, tá sa však okamžite rozvetvila na mnoho ďalších, navzájom rovnobežných, medzi ktorými sa kde-tu mihla výhybka. Tesne pred opustením priestoru stanice sa koľaje opäť zbehli do jedinej, po ktorej potom mohol vlak odísť.

Miško sa na svoje dielo nevedel dovynadávať, možno aj preto si nevšimol svojho mladšieho brata, ako nerozvážne položil na koľajnice vlakovú súpravu. Čo však bolo oveľa horšie, spustil zdroj napätia a v tom momente sa celá súprava pohla a každou sekundou sa blížila k stanici. V tej chvíli Miško precitol a celý zdesený spanikáril. Problém bol totiž v tom, že kvôli toľkému množstvu výhybiek na stanici hrozilo vykoľajenie celej súpravy. A čo keď sa nejako poškodí? To bude prúser jak mraky a navyše sa bude jeho mama hnevať. Neostávalo mu nič iné, ako prehodiť čo najmenej výhybiek, aby stihol zabrániť katastrofe. Keby mohol, určite by požiadal o pomoc vás. Schválne, vedeli by ste mu pomôcť?

²Tak napr. „1, a, a, 0“; $|a| = 2$ popisuje postupnosť šiestich kameňov, pričom (zľava) prvý má farbu 1, potom nasledujú dva rovnaké úseky, každý pozostávajúci z dvoch kameňov, a posledný kameň má farbu 0

³čomu sa jeho mama veľmi potešila

Úloha: Situáciu z Miškovej izby si pre jednoduchosť opíšeme nasledovne: je daných N rovnobežných nekonečne dlhých koľají očíslovaných od 1 po N , medzi ktorými je K výhybiek očíslovaných od 1 po K . Každá výhybka vedie medzi dvoma susednými koľajami. Žiadne dve výhybky nezačínajú na tej istej x -ovej súradnici a každá výhybka má dĺžku 1.

Stanica potom predstavuje miesto ohraničené výhybkami 1 a K , pričom tieto výhybky tiež patria do stanice. Výhybka vedúca z koľaje a na koľaj b sa môže nachádzať v dvoch stavoch. Hovoríme, že je „vypnutá“, keď vlak na úseku, kde sa výhybka nachádza, môže prechádzať po koľajach a , b rovno, nemôže však prejsť po výhybke z koľaje a na koľaj b . Keď je výhybka „zapnutá“, vlak môže prejsť po výhybke z koľaje a na koľaj b , ale nemôže prechádzať rovno. V žiadnom prípade však vlak nemôže prejsť z koľaje b na koľaj a .

Na vstupe sú najprv dané prirodzené čísla N, K, p, q . Číslo N teda udáva počet koľají a K počet výhybiek. Vlak vchádza do stanice po koľaji p a musí odísť po koľaji q . Ďalej nasleduje popis výhybiek. i -ta výhybka je popísaná číslami x_i, a_i, b_i, s_i , čo znamená, že výhybka vychádza z koľaje a_i na súradnici x_i a prichádza na koľaj b_i na súradnici $x_i + 1$. Výhybka je v „vypnutá“, ak $s_i = 0$, alebo zapnutá, keď $s_i = 1$. Váš program má vypísať minimálny počet výhybiek, aký treba prehodiť, aby vlak mohol prejsť stanicou. Môžete predpokladať, že výhybky sa dajú vždy prehodiť tak, aby vlak mohol prejsť stanicou.

Príklad:

Vstup:

$N = 4, K = 10, p = 3, q = 2$

1.0 3 2 0	6.5 3 2 0
2.0 3 4 1	9.0 2 1 1
3.0 2 1 1	9.5 4 3 1
4.0 2 3 1	11.0 3 2 0
5.0 4 3 1	13.0 1 2 1

Výstup:

Miško musí prehodiť 1 výhybku.

(Pozn: Treba prehodiť výhybku č. 6.)

4. O farebných trojuholníkoch

Janka sedí v škole už šiestu hodinu. Hádám už aj vyhladla. Ale učitelia do nej len hustia a hustia. Už ju to nebaví, a tak si z nudy začala kresliť. Najprv nakreslila na papier N malých bodiek. Aby bodkám nebolo smutno, tak ich očíslovala číslami od 1 po N . Potom zobrala červenú a modrú farbičku a začala ich spájať. Každé dve malé bodky spojila buď červenou alebo modrou úsečkou.

Miško sa pozrel na Jankin obrázok. Napadlo ho, že by mohol spočítať, koľko je na ňom takých trojuholníkov, ktoré majú strany rovnakej farby, či už červenej alebo modrej. Raz mu vyšlo niečo a potom niečo iné. Už mu akosi dochádza trpezlivosť. Pomôžte mu!

Úloha: Dané je N – počet bodiek a $\frac{N(N-1)}{2}$ trojíc čísel x, y, f , kde x a y sú čísla dvoch rôznych bodiek a f je farba úsečky, ktorou sú spojené. Každá dvojica bodiek je daná práve raz. Napíšte program, ktorý zistí, koľko je na takto zadanom obrázku jednofarebných trojuholníkov.

Príklad:

Vstup:

$N = 5$

1 2 červená	2 4 modrá
1 3 červená	2 5 červená
1 4 modrá	3 4 modrá
1 5 červená	3 5 červená
2 3 modrá	4 5 modrá

Výstup:

Sú tam 3 jednofarebné trojuholníky.

5. O čiernych krabičkách III

Neviem, či vám niečo hovorí názov IBM. Väčšine z vás asi nie. Len tým pár dobre informovaným zasvietili očička a s posvätnou úctou zašepkali: „*Institute of Black Magic*“. A tí, čo vedia, vám o nej aj čo-to povedia, keď sa ich opýtate. Napríklad by ste sa mohli dozvedieť, že v Inštitúte čiernej mágie (ďalej len IBM) majú celé oddelenie, kde za použitia čiernej mágie vyrábajú čierne krabičky. Napríklad aj tie, s ktorými ste sa mohli stretnúť v prvých dvoch kolách.

IBM dostali objednávku na nový typ čiernej krabičky a mali by ho mať hotový do 11. marca. Za použitia čiernej mágie by takú krabičku hravo zostrojili, lenže... Lenže nik si nevie ani len predstaviť, ako by mala vyzeráť.

Požiadavky klienta (výskumníčky Janky⁴) nie sú zďaleka zanedbateľné. Niektorí ľudia, ktorých skúma, sa priatelia, iní nie. Občas sa niektorí spriatelia alebo pohádajú, no a Janka to popri všetkých výskumoch jednoducho nezvláda všetko držať v hlave, tak si na to objednala čiernu krabičku.

Pamäť typu Čierna skrinka v3.0 by mala podporovať nasledovné príkazy:

- **procedure Init(N)**; – vyprázdni pamäť čiernej skrinky a nastaví si počet pamätaných ľudí na N. Pre jednoduchosť ich má Janka očíslovaných číslami od 1 do N. Na začiatku sa nikto nepriatelí s nikým.
- **procedure AddFriends(x,y)**; – ľudia x a y sa odteraz priatelia. (Ak sa už priatelili, nič sa nedeje.)
- **procedure DeleteFriends(x,y)**; – ľudia x a y sa odteraz nepriatelia. (Ak sa ani predtým nepriatelili, nič sa nedeje.)
- **procedure AreFriends(x,y)**; – Vypíše **ANO**, ak sa ľudia x a y práve priatelia a **NIE** inak.
- **procedure EnumFriends(x)**; – Vypíše (v ľubovoľnom poradí) čísla všetkých priateľov človeka x.

Úloha: Napíšte program, ktorý sa bude správať ako táto čierna krabička, teda bude čítať vstup od užívateľa a bude obsahovať vyššie uvedené procedúry. Formát vstupu si môžete navrhnuť tak, aby sa vám ľahko čítal.

Samozrejme je nutné, aby volanie každej z procedúr trvalo čo najkratšie. (Napríklad čas behu procedúry **EnumFriends** by nemusel závisieť od počtu všetkých ľudí, iba od počtu priateľov daného človeka) Pamätajte, že čím lepšiu krabičku navrhnete, tým efektívnejšiu pamäť z nej potom v IBM vyvinú a tým vám bude Janka vďačnejšia.

Príklad:

Vstup:	Výstup:
Init(3)	
EnumFriends(1)	(nikto)
AddFriends(1,2)	
AddFriends(1,3)	
DeleteFriends(2,3)	
DeleteFriends(1,3)	
AreFriends(1,3)	nie
EnumFriends(2)	1

⁴Pre tých, čo ju nepoznáte – Janka je biela myška, žijúca v kletke v jednom laboratóriu. Skúma a cvičí si ľudí okolo seba, medzi jej hlavné úspechy patrí, že ich dokázala vycvičiť, aby jej pravidelne dávali syr.