
Hodnocení soutěžních úloh

Kategorie žáci

Soutěž dětí a mládeže v programování – 22. ročník
Krajské kolo 2007/2008
18. a 19. dubna 2008

Olympijské logo

Koeficient 1

Vytvořte program, který zobrazí logo olympijských her, tedy pět vzájemně propletených kruhů dle níže uvedeného obrázku. Není nutné, aby místa, kde se kruhy překrývají, byla zobrazena přesně podle vzoru, ale čím více se váš výtvar bude vzoru podobat, tím větší bodové ohodnocení získáte.

Dostanete-li zadání jen jako černobílý výtisk, poradíme vám barvy jednotlivých kruhů – horní řada zleva: modrá, černá, červená; dolní řada zleva: žlutá, zelená.



Hodnocení

funkčnost	2 body	zobrazení pěti mezikruží na správných pozicích
	2 body	barvy odpovídají vzoru
	2 body	správné zobrazení překryvů dle vzorového loga (žádné mezikruží nesmí být opticky navrchu)
efektivita	1 bod	tloušťka mezikruží je větší než 1 pixel
	1 bod	logo není nepřiměřeně malé (=je větší než cca 100px na šířku)
	1 bod	logo není zubaté (=nejedná se o nekvalitní zvětšení malého obrázku)
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...

Sčítání čísel zapsaných římskými číslicemi

Koeficient 1

Náš starý známý strýček Pompo má kromě jiných zálib i zálibu v historii, protože ale jeho znalosti historie nejsou ještě tak dobré, potřeboval by od vás pomoci. Radost by mu udělal program, který dokáže sčítat čísla zapsaná římskými číslicemi.

Napište program, který dokáže sčítat po sobě jdoucí čísla zapsaná římskými číslicemi. Zadávání jednotlivých čísel se bude provádět z klávesnice. Po zadání prvních dvou čísel program ihned zobrazí jejich součet a bude připraven přičíst k výsledku další zadané číslo. Součet musí program zobrazit jako klasické číslo, zobrazíte-li součet i pomocí římských číslic, dostanete body navíc. Ukončení zadávání provede strýček Pompo zadáním slova „KONEC“. (Je na vás, zda bude Pompo mačkat tlačítko s nápisem „KONEC“, a nebo napíše slovo „KONEC“ do řádku pro zadávání čísel.)

Přestože zápisů čísel římskými číslicemi existuje několik, budeme považovat za správný pouze ten následující.

Číslo 3000 bude pro nás to nejvyšší možné. Římané sice uměli počítat i s většími čísly, ale Pompovi to bude stačit. Římané zapisovali svá čísla pomocí písmen, která vždy symbolizují jeden přesný počet. Zápis čísla začíná vždy písmenem s nejvyšší hodnotou, k tomuto písmenu se pak přiřazují písmena další, která jeho hodnotu zvyšují, všimněte si v následující tabulce čísel 4, 9, 14, 17, 60, 112, atd. Stačí tedy číst čísla zleva doprava a hodnoty jednotlivých písmen sčítat, tak nám vyjde výsledné číslo. Program bude kontrolovat zadané číslo a v případě chybného zápisu vyhlásí chybu.

Použité symboly jsou: I = 1, V = 5, X = 10, L = 50, C = 100, D = 500, M = 1000.

Příklady zápisu:

1 = I	6 = VI	11 = XI	16 = XVI	21 = XXI	40 = XXXX
2 = II	7 = VII	12 = XII	17 = XVII	24 = XXIIII	43 = XXXXIII
3 = III	8 = VIII	13 = XIII	18 = XVIII	26 = XXVI	45 = XXXXV
4 = IIII	9 = VIIII	14 = XIIII	19 = XVIIII	29 = XXVIIII	49 = XXXXVIIII
5 = V	10 = X	15 = XV	20 = XX	30 = XXX	50 = L
55 = LV	67 = LXVII	103 = CIII	129 = CXXVIIII	400 = CCCC	
59 = LVIIII	70 = LXX	112 = CXII	130 = CXXX	450 = CCCCL	
60 = LX	80 = LXXX	115 = CXV	150 = CL	462 = CCCCLXII	
63 = LXIII	90 = LXXXX	120 = CXX	200 = CC	500 = D	
65 = LXV	100 = C	124 = CXXIIII	233 = CCXXXIIII	501 = DI	
574 = DLXXIIII	700 = DCC	900 = DCCCC	1300 = MCCC		
599 = DXCIVIIII	751 = DCCLI	937 = DCCCCXXVII	1631 = MDCXXXI		
600 = DC	790 = DCCLXXXX	953 = DCCCCLIII	2200 = MMCC		
650 = DCL	800 = DCCC	978 = DCCCCLXXVII	2750 = MMDCCCL		
681 = DCLXXXI	825 = DCCCXXV	1000 = M	3000 = MMM		

Hodnocení

Program vyžaduje vstup a kontroluje správnost zadání	0.25 bodu	Program přijme vstup IIII
	0.25 bodu	Program přijme vstup MDC
	0.5 bodu	Program vypíše chybu pro vstup AAA
	0.5 bodu	Program vypíše chybu pro vstup VXL
	0.5 bodu	Program vypíše chybu pro vstup MMMM
Program umí správně sečíst dvě zadána čísla	0.5 bodu	III + II = 5 (V)
	0.5 bodu	VI + IIII = 10 (X)
	0.5 bodu	LX + XX = 80 (LXXX)
	0.5 bodu	MDCCC + DC = 2400 (MMCCCC)
Program umí správně sečíst tři zadána čísla (první čtyři sčítání lze kontrolovat zároveň s předchozími čtyřmi)	0.5 bodu	III + II + CLV = 200 (CC)
	0.5 bodu	VI + IIII + MDXX = 1530 (MDXXX)
	0.5 bodu	LX + XX + MMI = 2081 (MMLXXXI)
	0.5 bodu	MDCCC + DC + CVI = 2506 (MMDVI)
	0.5 bodu	D + IIII + CVI = 610 (DCX)
	0.5 bodu	MMDC + CVIII + CCLXXXII = 3000 (MMM)
Další funkčnost	0.5 bodu	Program ohlásí chybu při součtu větším než 3000 (MM + MI)
	0.5 bodu	Program reaguje na zadání „KONEC“
	1 bod	Bonus bod za výpis výsledku v římských číslech
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...

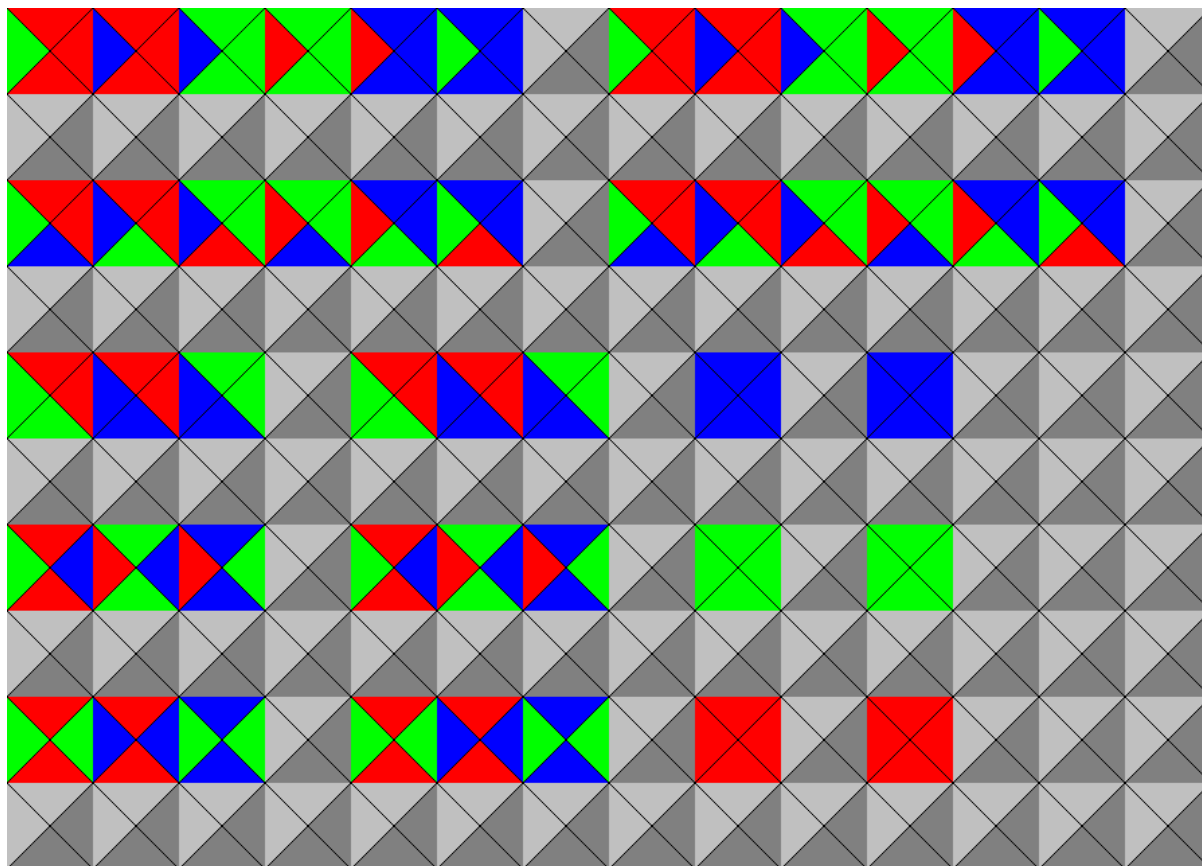
Quadromino

Koeficient 2

Jak jistě víte, strýček Pompo je dlouhodobým hráčem korespondenčního Quadromina. Naším úkolem je mu pomoci a vytvořit program na zobrazování rozehrané hry zadané v souboru.

Quadromino je desková stolní hra podobná hře Domino. Rozdíl je v možnosti přikládat kostku ke kostce ne ze dvou stran, ale ze čtyř. Stejně jako u Domina existují varianty pro jednoho, dva a více hráčů. Hraje se s kostkami – čtvercovými kameny, kde každá ze čtyřech stran má svoji barvu.

Obrázek 1. Ukázka různých druhů kostek

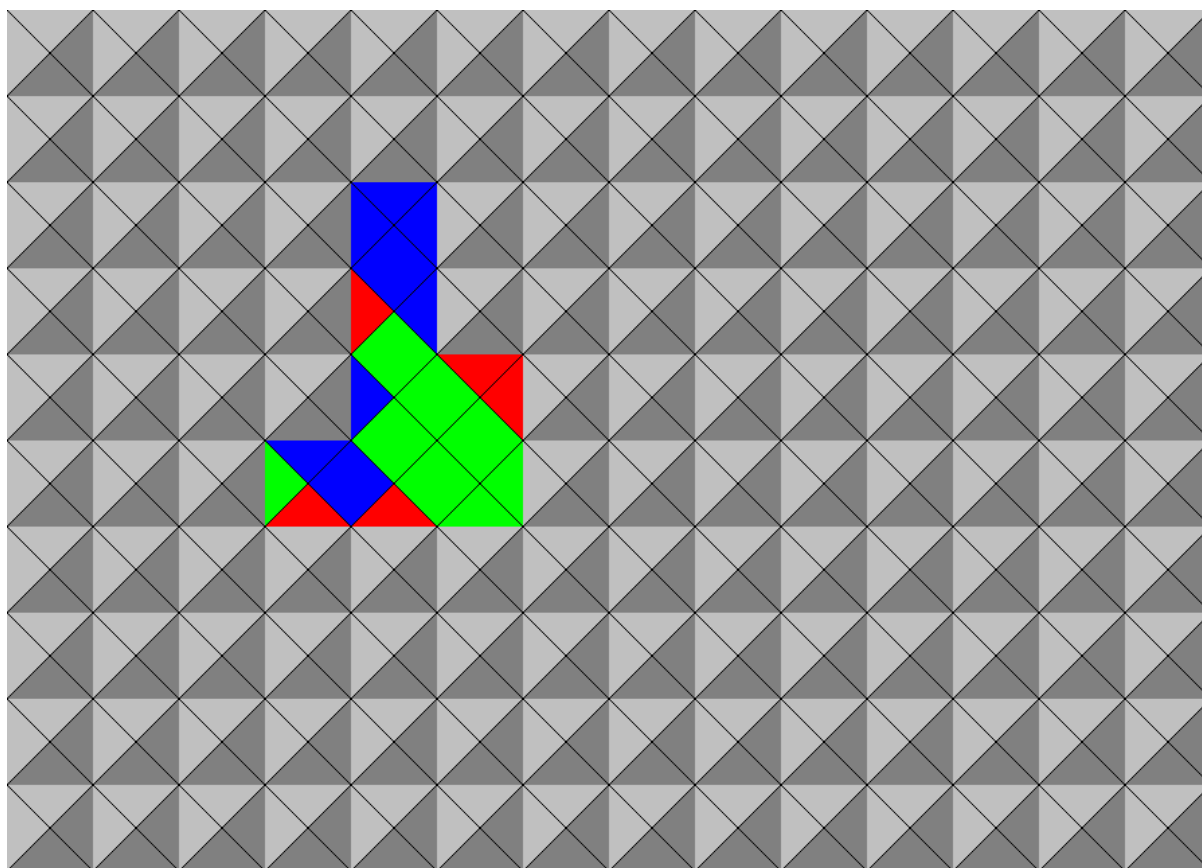


Barvy rozeznáváme tři: červenou = R, zelenou = G a modrou = B. Minimálně dvě strany proto musí mít shodnou barvu. Začíná se s jednou kostkou a rozehraná hra tedy obsahuje minimálně jednu kostku. Pokud je více kostek, každá musí sousedit alespoň s jednou další kostkou. Barvy přiléhajících stran sousedních kostek jsou vždy shodné, jak ukazuje obrázek 2.

Vytvořte program, který zobrazí rozehranou hru zadanou v souboru (soubor si přitom uživatel může vybrat). Dále přidejte kontrolu rozestavených kostek na správnost. Pro správně rozestavené kostky platí, že

- sousední kostky mají stejnou barvu a
- žádná kostka neleží osamoceně, kromě případu, kdy je jediná.

Obrázek 2. U přiléhajících kostek na sebe musí navazovat barvy



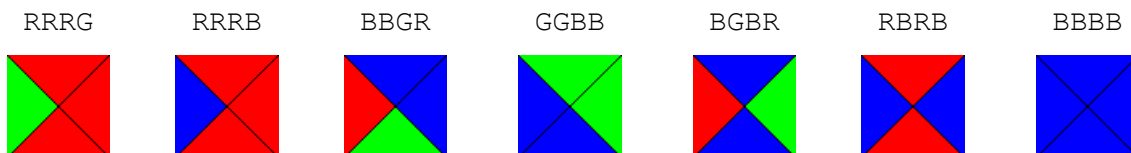
Popis formátu souboru

Soubor s popisem rozehrané hry je textový soubor. Každé pole je v něm zaznamenáno jako čtyři znaky. Každý řádek souboru odpovídá jednomu řádku rozehrané hry. Řádky mohou být odděleny pomocí znaků CR, LF nebo CR+LF.

Prázdné pole bez kostky jsou znaky ----.

Plné pole obsazené kostkou vypadá takto ????, kde místo každého otazníku si dosadíte písmeno R, G nebo B. První písmeno udává barvu na horní straně kostky, druhé je barva vpravo, třetí je barva dole a čtvrté je barva vlevo.

Příklady:



Kódy BGBR, GBRB, BRBG, RGBB jsou jedna kostka, jen různě otočená:



Kódy BBGR a BBRG jsou dvě různé kostky:



Následující soubor popisuje rozehranou hru zobrazenou na obrázku 2. Další příklady najdete v adresáři s testovacími soubory.



Hodnocení

funkčnost	1 bod	vstupní soubor lze zvolit komfortně
	2 body	program správně zobrazí soubor a.txt
	1 bod	program správně zobrazí soubor b.txt
	2 body	program správně zobrazí soubor c.txt
	0.5 bodu	program oznámí, že soubor d1.txt nevyhovuje pravidlům (izolovaná kostka)
	0.5 bodu	program oznámí, že soubor d2.txt nevyhovuje pravidlům (nenavazující kostka)
	1 bodu	program oznámí, že soubor d3.txt má špatný tvar
efektivita	1 bod	kontrola správného rozestavení kostek se provádí efektivně – kontroluje se shoda kostka1 s kostka2 a kostka2 s kostka1 se už neprovádí
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...

Body za zobrazení souboru se započítávají pouze v případě, kdy je soubor zobrazen do 10 sekund.

Loupežníci

Koeficient 3

Dvě známé loupežnické firmy Rumcajs, s radostí okrádám, a Lotrando, absolutní straka, spojily své síly a společně uloupily velkou kořist. Kořist se skládá z N předmětů, každý předmět má nějakou cenu c_i . Nyní se loupežníci musí o lup rozdělít. Protože ale chtějí i nadále spolupracovat, musí se rozdělít rovným dílem, tedy tak, aby oba dostali přesně polovinu celého lupu. Nemohou se ale dohodnout a už na sebe začínají vytahovat bambitky. Pomůžete jim?

Vaším cílem je napsat program, který na vstupu dostane číslo N , dále N cen jednotlivých předmětů lupu c_1, \dots, c_N a zjistí, zda je možné tyto předměty rozdělít do dvou skupin tak, aby se součet cen předmětů v jedné skupině rovnal součtu cen předmětů v druhé skupině. Žádný z předmětů už není možné dělit, musí být celý buď v jedné nebo druhé skupině. Pokud lze předměty rozdělít, program vypíše, jaké předměty jsou v jedné z těchto dvou skupin. Pokud řešení existuje několik, vypsat můžete libovolné z nich.

Pozor, musíte přesně dodržet popsany formát vstupu a výstupu, protože úloha se bude vyhodnocovat automaticky. Pokud popsany formát nedodržíte, neobdržíte žádné body.

Popis vstupu

Vstup je uložen v souboru `lup.in` v aktuálním adresáři. Na první řádce se nachází jediné číslo $1 \leq N \leq 20$. Na druhé řádce se nachází N přirozených čísel c_1, \dots, c_N – ceny předmětů lupu. Tato čísla jsou oddělena jednou mezerou, každé je větší rovno jedné a součet těchto N čísel je menší rovno 500000.

Řádky mohou být odděleny pomocí znaků CR, LF nebo CR+LF.

Popis výstupu

Výstup musíte vypsat do souboru `lup.out`. Pokud dané předměty nejdou rozdělít na dvě stejně cenné skupiny, musí výstupní soubor obsahovat jedinou řádku, na které se nachází číslo 0. Pokud dané předměty jdou rozdělít, musíte vypsat předměty z jedné skupiny. Výstupní soubor pak musí obsahovat dvě řádky. Na první musí být jediné číslo K – počet předmětů v jedné ze skupin. Na druhé řádce musí být mezerou oddělených K čísel – čísla předmětů ve skupině (počítáno od jedničky). Tato čísla předmětů mohou být vypsána v libovolném pořadí, ale žádné se samozřejmě nesmí opakovat.

Příklad 1.

Pro následující vstupní soubor `lup.in`

```
5
2 5 4 7 2
```

předměty rozdělít nelze. Obsah souboru `lup.out` proto bude

```
0
```

Příklad 2.

Pro vstupní soubor `lup.in`

```
6
7 1 2 3 4 5
```

existují čtyři správná rozdělení. Vypsát můžete libovolné z nich, předměty mohou být vypsány v libovolném pořadí. Soubor `lup.out` může mít jeden z následujících obsahů

```
2
1 5 (7+4=11)
```

```
3
1 2 4 (7+1+3=11)
```

```
3
3 5 6 (2+4+5=11)
```

```
4
2 3 4 6 (1+2+3+5=11)
```


Hodnocení

Za tuto úlohu můžete udělit až deset bodů. Až jeden bod udělte za úpravu zdrojového kódu (pochopitelnost, pojmenování proměnných, odsazení, ...). Zbýlých devět bodů udělte za funkčnost a efektivitu dle následující tabulky:

funkčnost a efektivita	1.5 bodu	program správně vyřeší vstupy z1a a z1b
	1.5 bodu	program správně vyřeší vstupy z2a a z2b
	1.5 bodu	program správně vyřeší vstupy z3a a z3b
	1.5 bodu	program správně vyřeší vstupy z4a a z4b
	1.5 bodu	program správně vyřeší vstupy z5a a z5b
	1.5 bodu	program správně vyřeší vstupy z6a a z6b
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...

Pokud program vyřeší pouze jeden vstup ze dvojice, nepřiděluje žádné body. Na každý vstup má program 5 vteřin procesorového času. Pokud běží déle, je to stejné, jako by vstup nevyřešil (ohledně měření času viz dále). Testovací vstupy nesmíte dát soutěžícím v průběhu soutěže ani po jejím konci k dispozici.

Protože správných řešení je několik, správnost řešení je třeba testovat dodaným programem. Pro windows jsou tyto programy v archivu `loupeznici_win.zip`, pro linux je v archivu `loupeznici_linux.tgz`. Tento archiv obsahuje také všechny testovací vstupy a zdrojové kódy obou testovacích programů. Programy pro testování řešení jsou dva: `otestuj` a `spust_a_otestuj`. Testovací programy fungují pouze s dodanými testovacími vstupy.

Spust_a_otestuj: testování program, který i spouští řešení a měří čas. Nakopírujte vstupní data a program `spust_a_otestuj` do adresáře, kde je spustitelný soubor `reseni` od soutěže. Programu `spust_a_otestuj` dejte jako první parametr název vstupního souboru bez koncovky a jako druhý parametr příkaz, kterým se spustí řešení od soutěže (tj. třeba `spust_a_otestuj.exe m4a loupeznici.exe` na windows a `./spust_a_otestuj z5b ./loupeznici` na linuxu). Program `spust_a_otestuj` zkopíruje vstupní soubor do souboru `lup.in`, smaže `lup.out`, pokud existuje, dále spustí řešení, změří mu čas (a ukončí ho, pokud běží déle než 5 sekund) a nakonec otestuje, zda soubor `lup.out`, který mělo řešení vytvořit, obsahuje správný výsledek. Výsledek vypíše na konzoli.

Otestuj: testovací program, který pouze testuje správnost řešení. Pokud by vám předchozí způsob z nějakého důvodu nevyhovoval, máte k dispozici ještě testovací program `otestuj`. Tento program pouze načte soubory `lup.in` a `lup.out` z aktuálního adresáře a zjistí, zda je `lup.out` správné řešení `lup.in`. Celé testování pak musí vypadat tak, že vy zkopírujete vstupní soubor do souboru `lup.in`, odstraníte soubor `lup.out`, pokud existuje, dále ručně spustíte řešení soutěže, změříte mu čas, a pokud běželo méně než 5 sekund a vytvořilo soubor `lup.out`, spustíte `otestuj`, abyste zjistili, zda bylo vyprodukované řešení správné.