Zadání soutěžních úloh

Kategorie mládež

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

Úlohy můžete řešit v libovolném pořadí a samozřejmě je nemusíte vyřešit všechny. Za každou úlohu můžete dostat maximálně 10 bodů, z nichž je většinou 9 bodů vyhrazeno na ohodnocení funkčnosti programu, jeho shody se zadáním a efektivity a jeden bod na dokumentaci a přehlednost zdrojového kódu. Body získané za každou úlohu se ještě násobí koeficientem, který odráží složitost úlohy.

Na řešení úlohy máte 4 hodiny čistého času.

Před zahájením soutěže vám pořadatel oznámí, kde najdete testovací soubory a vzorová řešení úloh.

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 výtvor 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.

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. Čísla jsou uložena v textovém souboru, který si uživatel může vybrat. Na každé řádce souboru je pouze jedno číslo zapsané římskými číslicemi. Váš program tento soubor načte a zobrazí celkový součet zapsaný jak arabskými tak římskými číslicemi.

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

Číslo 5000 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 následujiící 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 = CCXXXIII  501 = DI

574 = DLXXIIII  700 = DCC       900 = DCCCC             1300 = MCCC
599 = DXCIVIIII 751 = DCCLI     937 = DCCCCXXXVII       1631 = MDCXXXI
600 = DC        790 = DCCLXXXX  953 = DCCCCLIII         2200 = MMCC
650 = DCL       800 = DCCC      978 = DCCCCLXXVIII      2750 = MMDCCL
681 = DCLXXXI   825 = DCCCXXV   1000 = M                3000 = MMM
```

Interpret ŠVG

Koeficient 3

Napište program, který dokáže na obrazovce vykreslovat obrázky ve formátu ŠVG (**Š**mrncovní **V**obrázky a **G**rafika). V programu postupně naimplementujte následující funkce.

1. Výběr souboru s obrázkem k vykreslení.
2. Vykreslení obrázku přes celou plochu okna programu.
3. Zvětšování/zmenšování obrázku.
4. Posouvání obrázku.

Formát ŠVG

ŠVG je vektorový grafický formát. To znamená, že obrázek je složen ze základních geometrických objektů, u kterých lze určovat jejich pozici, rozměry a další vlastnosti jako barvu a výplň.

Formát ŠVG používá syntaxi XML. Celý obrázek je uzavřen v elementu `svg`. Uvnitř něj se pak nachází elementy popisující jednotlivé objekty, ze kterých se skládá celý obrázek. Obrázek se vykresluje vždy na bílé pozadí. Jednotlivé objekty, ze kterých se obrázek skládá, se vykreslují ve stejném pořadí v jakém jsou uvedeny v souboru.

Každý obrázek se vykresluje na plochu, jejíž rozměry jsou určeny pomocí atributů `width` (šířka) a `height` (výška). Rozměry mohou být určeny jako libovolné kladné desetinné číslo. Váš program musí podle potřeby tyto „uživatelské“ souřadnice přepočítávat tak, aby se obrázek vešel na obrazovku. Proporce obrázku přitom musí být zachovány.

Příklad 1. Ukázka kostry ŠVG obrázku

```
<?xml version="1.0" encoding="utf-8"?>
<svg width="320" height="200">
  …
</svg>
```

Obrázek se může skládat z následujících objektů:

čára

```
<line x1="…" y1="…" x2="…" y2="…"/>
```

Čára je určena souřadnicemi svého počátku (x1,y1) a konce (x2,y2).

obdélník

```
<rect x1="…" y1="…" x2="…" y2="…"/>
```

Obdélník je určen souřadnicemi diagonálně protilehlých rohů se souřadnicemi (x1,y1) a (x2,y2).

obdélník s oblými rohy

```
<rect x1="…" y1="…" x2="…" y2="…" rx="…" ry="…"/>
```

Obdélník je určen souřadnicemi diagonálně protilehlých rohů se souřadnicemi (x1,y1) a (x2,y2). Jeho rohy nejsou pravoúhlé, ale jsou nahrazeny odpovídající čtvrtinovou výsečí elipsy, jejíž poloměr v osách X a Y je určen atributy `rx` a `ry`.

kružnice

```
<circle cx="…" cy="…" r="…"/>
```

Kružnice je určena souřadnicemi svého středu (cx,cy) a svým poloměrem (r).

elipsa

```
<ellipse cx="…" cy="…" rx="…" ry="…"/>
```

Elipsa je určena souřadnicemi svého středu (cx,cy) a poloměrem hlavní (ry) a vedlejší (rx) osy.

U každého obrázku je možné pomocí následujících atributů určit barvu obrysu, sílu obrysu a barvu výplně:

`color` Barva obrysu objektu. Barva se určuje podobně jako v jazyce HTML pomocí zápisu `#RRGGBB`, kde `RR` je intenzita červené složky, `GG` je intenzita zelené složky a `BB` je intenzita modré složky v barevném modelu RGB zapsané v šestnáctkové soustavě.

Příklady některých barev: `#000000` (černá), `#FFFFFF` (bílá), `#FF0000` (červená), `#FFFF000` (žlutá).

Pokud barva není určena, předpokládá se černá.

`fill` Barva výplně objektu (netýká se čár). Barva se určuje stejně jako pro atribut `color`.

Pokud barva výplně není určena, předpokládá se bílá.

`stroke` Síla čáry obrysu zadaná jako destinné číslo. Pokud není síla určena, předpokládá se 1 (v „uživatelských“ souřadnicích).

Objekty, ze kterých se obrázek skládá, je možné sdružovat pomocí elementu `g` a na něm nastavit pro všechny sdružené objekty barvu obrysu, sílu obrysu a barvu výplně. Pokud je však některý z těchto parametrů nastaven přímo na daném grafickém objektu, má přednost před definicí zděděnou z elementu `g`. Elementy `g` je možné do sebe navzájem zanořovat.

Příklad 2. Ukázka jednoduchého obrázku a jeho zobrazení

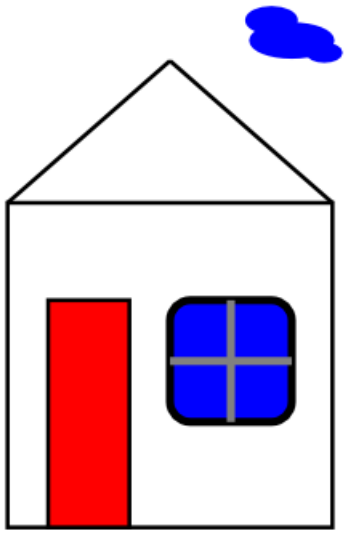
```
<?xml version="1.0" encoding="utf-8"?>
<svg width="100" height="150">
  <!-- HlavnÄ ÄŤÄ˘st domeÄŤku -->
  <rect x1="10" y1="10" x2="90" y2="90" color="#000000"/>

  <!-- StĹ™echa -->
  <line x1="10" y1="90" x2="50" y2="125" color="#000000"/>
  <line x1="90" y1="90" x2="50" y2="125" color="#000000"/>

  <!-- DveĹ™e -->
  <rect x1="20" y1="10" x2="40" y2="66" color="#000000" fill="#FF0000"/>

  <!-- OkÄ©nko -->
 <rect x1="50" y1="36" x2="80" y2="66" rx="5" ry="5" color="#000000" fill="#0000FF"
 stroke="2"/>
  <line x1="65" y1="36" x2="65" y2="66" stroke="2" color="#808080"/>
  <line x1="50" y1="51" x2="80" y2="51" stroke="2" color="#808080"/>

  <!-- MrÄ˘ÄŤek -->
  <g color="#0000FF" fill="#0000FF">
    <ellipse cx="80" cy="130" rx="10" ry="4"/>
    <ellipse cx="75" cy="135" rx="6" ry="3"/>
    <ellipse cx="88" cy="127" rx="4" ry="2"/>
  </g>
</svg>
```



Další ukázky obrázků najdete v adresáři se zadáním úloh.

Loupežníci

Koeficient 3

Dvě známé loupežnické firmy Rumcajs, **s** **r**adostí **o**krádám, a Lotrando, **a**bsolutní **s**traka, 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 rozdělit. Protože ale chtějí i nadále spolupracovat, musí se rozdělit 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,...,c_N a zjistí, zda je možné tyto předměty rozdělit 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 rozdělit, program vypíše, jaké předměty jsou v jedné z těchto dvou skupin. Pokud řešení existuje několik, vypsat můžete libovolné z nich.

Pozor, musíte přesně dodržet popsaný formát vstupu a výstupu, protože úloha se bude vyhodnocovat automaticky. Pokud popsaný 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<=N<=500. Na druhé řádce se nachází N přirozených čísel c_1,...,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 rozdělit na dvě stejně cenné skupiny, musí výstupní soubor obsahovat jedinou řádku, na které se nachází číslo 0. Pokud dané předměty jdou rozdělit, 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 3.

Pro následující vstupní soubor `lup.in`

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

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

```
0
```

Příklad 4.

Pro vstupní soubor `lup.in`

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

existují čtyři správná rozdělení. Vypsat můžete libovolné z nich, předměty mohou být vypsané 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)
```