
Hodnocení soutěžních úloh

Kategorie mládež

Soutěž v programování – 24. ročník
Krajské kolo 2009/2010
15. až 17. dubna 2010

Superciferný součet

Koeficient 1

Vaší úlohou je vytvořit program, který spočítá superciferný součet zadaného čísla.

Ciferný součet je číslo, které je součtem všech číslic zadaného čísla. Superciferný součet je takové číslo, které vznikne opakováním operace ciferný součet tak dlouho, dokud není výsledek menší než 10.

Příklad 1.

Zadané číslo = 99 999 999 999

Ciferný součet = $9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 = 99$

Protože výsledek není menší než 10, opakujeme operaci

Ciferný součet = $9 + 9 = 18$

Protože výsledek není menší než 10, opakujeme operaci

Ciferný součet = $1 + 8 = 9$

9 je výsledný superciferný součet

Příklad 2.

Zadané číslo = 1 234 567 890

Ciferný součet = $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 0 = 45$

Protože výsledek není menší než 10, opakujeme operaci

Ciferný součet = $4 + 5 = 9$

9 je výsledný superciferný součet

Zadané číslo se čte z textového souboru zvoleného uživatelem. Jiné znaky než číslice při čtení souboru ignorujte.

Vstupní soubor může být velmi dlouhý. Počet číslic ale nepřesáhne 238 609 294.

Výstupem programu je výsledný superciferný součet zobrazený na obrazovce.

Hodnocení

Funkčnost	1 bod	Program lze spustit
	1 bod	Lze zadat vstupní soubor
	1 bod	Lze zadat vstupní soubor komfortně -> procházení adresářů -> nemusí se vypisovat ručně
	1 bod	Program spočítá <code>test1.txt</code> (výsledek je 3) do 2 s (malý soubor – lze spočítat z hlavy)
	1 bod	Spočítaný ciferný součet je na výstupu jasný a zřetelný
	1 bod	Program spočítá <code>test2.txt</code> (výsledek je 9) do 5 s (soubor velký cca 1kB, obsahuje různé znaky)
	2 body	Program spočítá <code>test3.txt</code> (výsledek je 7) do 10 s (soubor velký cca 500MB, obsahuje různé znaky, počet číslic ale nepřesáhne 238 609 294)
	1 bod	Program počítá vše do 1 s nebo ukazuje průběh výpočtu
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...

Závorky

Koeficient 1

Napište program, který dostane jako parametr vstupní soubor. V souboru je velké množství závorek různých typů:

- kulaté – (...)
- hranaté – [...]
- složené – { ... }
- lomené – < ... >

Kromě závorek se v souboru mohou vyskytovat i další znaky, které můžete ignorovat.

Zjistěte, zda je uzávorkování správné. Uzávorkování je správné když:

- první závorka je levá a existuje k ní pravá závorka stejného typu;
- vnitřek mezi těmito závorkami je správně uzávorkován;
- zbytek za pravou závorkou je správně uzávorkován.

Program vypíše „OK“, je-li uzávorkování správné a „Chyba“, není-li uzávorkování správné.

Příklad 3.

```
(<{><>>){<><>} → OK  
<{><>)>{<><>} → Chyba  
<{><>>{<><>} → Chyba
```

Zkuste program napsat tak, aby fungoval rychle i pro velké soubory (řádově stovky miliónů závorek).

Hodnocení

Funkčnost	1 bod	Program lze spustit a soubor <code>nic.txt</code> lze vyhodnotit (ne nutně správně) a program nehavaruje
	1 bod	Vstupní soubor lze zadat komfortně -> procházení adresářů -> nemusí se vypisovat ručně
	1 bod	Správně vyhodnotí všechny soubory <code>zadani1.txt</code> , <code>zadani2.txt</code> a <code>zadani3.txt</code>
	1 bod	Správně vyhodnotí oba soubory <code>TestA_ERR.txt</code> a <code>TestA_OK.txt</code> , jinak nic
	1 bod	Správně vyhodnotí soubor <code>TestB_OK.txt</code>
	1 bod	Správně vyhodnotí soubor <code>TestC_OK.txt</code>
	1 bod	Program vyhodnotí soubor 200MB <code>Test_big_OK</code> do 8 minut
	1 bod	Program vyhodnotí soubor 200MB <code>Test_big_ERR</code> do 8 minut
	1 bod	Program počítá vše do jedné minuty nebo ukazuje průběh výpočtu (např. pomocí procent, progressbaru, ...)
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...

Translácie

Koeficient 2

Obyvatelé planety Translácie se vždy živili tím, že překládali stránky nalezené vyhledávači Túdle a Blink do jazyků všech ostatních galaktických národů. Kvůli nenadálé ekologické katastrofě však na Transláci vyhynuly všechny vzácné babylonské rybky, které se o překládání staraly. Pomozte se obyvatelům Translácie zachránit. Mají dobré překladače, ale ti umějí vždy překládat jen z jednoho jazyka.

Napište program, který pozná v jakém jazyce je zapsán vstupní soubor, aby mohl být odeslán správnému překladači. Jazyk celého souboru určete podle jazyka převládajícího u nejvíce slov. Aby program mohl určit příslušnost slova k nějakému jazyku, bude mít na vstupu i slovníky jednotlivých jazyků. Takových slovníků může být maximálně 16.

Za slovo se považuje souvislá posloupnost složená ze znaků A–Z (na velikosti písmen nezáleží). Jakýkoliv jiný znak je považován za oddělovač slov.

Soubor se slovníkem je textový soubor, kde je na každé řádce jedno slovo. Jazyk slovníku odpovídá jménu souboru.

Příklad 4. Ukázka určení jazyka

Vstupní soubor:

```
Bezi liska k taboru  
nese pytel zazvoru.
```

Soubor slovníku cesky:

```
ahoj  
amater  
...  
liska  
...  
pytel  
...  
zazvoru  
...
```

Soubor slovníku anglicky:

```
abort  
after  
...  
fox  
...  
ginger  
...
```

Očekávaný výstup programu:

```
Vstupní text je cesky
```

Hodnocení

Funkčnost	1 bod	Program lze spustit a lze mu předat parametry – vstupní soubor a soubory se slovníky
	1 bod	Program správně určí jazyk souboru <code>ustava.txt</code> při použití dvou slovníků <code>cesky</code> a <code>slovensky</code> (vstup je česky)
	1 bod	Program oznámí nemožnost zjištění jazyka, pokud se slova ze souboru <code>nesmysly.txt</code> nevyskytují ani v jednom slovníku (testovat lze pro libovolné dva slovníky)
	1 bod	Program si poradí se situací, kdy vstup <code>nastejno.txt</code> obsahuje stejně slov ze dvou různých jazyků (použijte slovníky <code>cesky</code> a <code>anglicky</code>)
	1 bod	Program správně určí jazyk souboru <code>xml.txt</code> jako francouzštinu pro 3 slovníky (<code>nemecky</code> , <code>anglicky</code> a <code>francouzsky</code>)
	1 bod	Program správně určí jazyk souboru <code>xml.txt</code> jako francouzštinu pro 4 slovníky (<code>cesky</code> , <code>nemecky</code> , <code>anglicky</code> a <code>francouzsky</code>)
	1 bod	Program správně určí jazyk souboru <code>xml.txt</code> jako francouzštinu pro 5 slovníků (<code>cesky</code> , <code>slovensky</code> , <code>nemecky</code> , <code>anglicky</code> a <code>francouzsky</code>)
	2 body	Program okamžitě (do cca 3 sekund) a správně určí jazyk jako češtinu u velkého vstupu <code>velky.txt</code> při použití slovníků <code>cesky</code> a <code>anglicky</code>
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...

PEXESO

Koeficient 2

Napište program pro hru PEXESO pro jednoho hráče.

Velikost hracího pole je 8×8 nebo 4×4 hracích karet. Sady obrázků pro hrací karty jsou umístěny v podadresáři `obrazky`. Každá sada má určený název, který je shodný s názvem podadresáře. V podadresáři jsou už obrázky očíslovány od 01 až do 32 a uloženy v souboru s příponou `.png`. Tyto obrázky definují líc hracích karet. Obrázek `00.png` vždy definuje rub hracích karet.

Například sada „vlajky“ je umístěna v adresáři `./obrazky/vlajky` a jednotlivé obrázky mají název `01.png`, `02.png`, ..., `32.png`.

Pro velikost hracího pole 4×4 použijte pouze obrázky `01.png` až `08.png`.

Po spuštění programu PEXESO umožněte uživateli zvolit si velikost hracího pole a sadu obrázků. Dále si uživatel zvolí, zda se na začátku hry zobrazí hrací karty rubem nebo lícem nahoru. V případě volby lícem nahoru se po 30 sekundách všechny karty otočí zpátky do stavu s rubem nahoru.

Zobrazte počítadlo všech tahů a počítadlo úspěšných tahů.

Tahem se rozumí kliknutí myši uživatelem na dvojici neotočených (rubem nahoru) hracích karet. Kliknutí na otočenou hrací kartu (lícem nahoru) je ignorováno.

Každé platné kliknutí se projeví otočením hrací karty. Úspěšným tahem se rozumí otočení dvojice stejných hracích karet a hrací karty zůstanou trvale otočené. Při neúspěšném tahu se karty vrátí po dvou sekundách zpět. Po každém tahu se aktualizují počítadla tahů.

Hra končí při úspěšném otočení všech hracích karet.

Nakonec zobrazte počty všech tahů a úspěšných tahů. Tah zobrazíte jako dvojici čísel hracích karet oddělených čárkou. Úspěšný tah bude mít za dvojicí čísel ještě znak `+` (plus). Pak program ukončete nebo nabídněte možnost hraní další hry.

Hodnocení

Funkčnost	1 bod	možnost zadání velikosti a sady hracích karet a způsob zobrazení – rub / líc
	1 bod	načtení a zobrazení hracích karet do hracího pole 4x4
	1 bod	načtení a zobrazení hracích karet do hracího pole 8x8
	0.5 bodu	zobrazení hracích karet lícem nahoru se po 30s vrátí zpět
	0.5 bodu	kliknutí na kartu s lícem se ignoruje
	1 bod	kliknutí na kartu s rubem a karta se otočí lícem nahoru
	1 bod	neúspěšný tah – karta se po cca 2 s otočí rubem nahoru
	0.5 bodu	kliknutí na třetí kartu během neúspěšného tahu se ignoruje
	1 bod	počítadlo průběžně zobrazuje správně počet všech a úspěšných tahů
	1 body	zobrazení všech tahů na konci hry
	0.5 bodu	zobrazení úspěšnosti tahů
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...

Domino

Koeficient 3

Napište program DOMINO, který načte z grafických souborů hrací kostky hry DOMINO. Určí jejich počty puntíků a uspořádá je tak, že nalezne nejdelší řetěz hracích kostek.

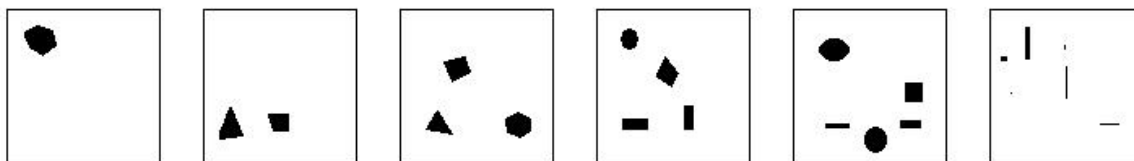
Po spuštění programu DOMINO umožněte uživateli si zvolit adresář, kde jsou grafické soubory ve formátu png (`kostka01.png`, `kostka02.png`, ...). Každý jednotlivý soubor reprezentuje hrací kostku hry domino. Soubor je definován jako obdélník, kde výška a délka obdélníku je v poměru 2:1 nebo 1:2. Plocha obdélníku je rozdělena na dva stejné čtverce ležící vedle sebe nebo nad sebou. Každý čtverec má rám o velikosti jednoho pixelu. Středová úsečka má tedy šířku dvou pixelů. Minimální rozměr hrací kostky je 50×100 pixelů a maximální 500×1000 pixelů včetně rámu.

V souborech jsou pouze dvě barvy: barva podkladu a barva puntíků. Rámy mají barvu puntíků.

Puntíky mohou mít různý tvar a všechny pixely jednoho puntíku spolu sousedí vodorovně nebo svisle, čili nestačí, pokud spolu sousedí pouze přes roh. Mezi rámem a puntíkem je minimálně jeden pixel. V každém čtverci musí být aspoň jeden puntík a nejvýše šest puntíků.

Puntíky jsou umístěny libovolně tak, že se vzájemně nedotýkají vodorovně nebo svisle, ale mohou se dotýkat přes roh jednoho pixelu.

Zde je ukázka tvarů a umístění puntíků ve čtverci:



Vášim úkolem je načíst hrací kostky a zjistit počty puntíků každé hrací kostky. Za to dostanete polovinu bodů. Výstup uložte do souboru `kostky.txt` tak, že na každém řádku bude jméno souboru hrací kostky bez přípony `png` a vedle ní počty puntíků ve formátu `x:y`, kde `x` je počet puntíků levého nebo horního čtverce a `y` počet puntíků na pravém či dolním čtverci. Jména souborů s kostkami musí být ve vzestupném abecedním pořadí.

Zbytek bodů dostanete za nalezení co nejdelšího řetězu hracích kostek takového, že se skládá z ležících kostek (dané kostky tedy můžete libovolně otočit tak, aby ležely) a sousedící kostky se dotýkají čtverci se stejným počtem puntíků. Pokud nejdelší řetěz hracích kostek naleznete, vypište jeho délku jako jediný řádek do souboru `retez.txt`.

Počet hracích kostek domina může být maximálně 50.

Hodnocení

Zkouší se na devíti vstupních adresářích (v kategorii žáci to jsou adresáře `zaci_1..zaci_9`, v kategorii mládež adresáře `mladez_1..mladez_9`), za každý může soutěžící dostat 0 bodů, 0.5 bodu nebo 1 bod.

Program soutěžících se spustí s daným adresářem a musí doběhnout do 10 vteřin.

Pokud program vytvoří soubor `kostky.txt`, který se přesně shoduje se vzorovým, dostane 0.5 bodu. (Porovnání dvou souborů podle obsahu lze provést ve Windows pomocí příkazu `fc`, v Linuxu pomocí příkazu `diff`.)

Pokud program vytvořil také soubor `retez.txt`, který se přesně shoduje se vzorovým, dostane dalších 0.5 bodu.

Pokud program nevytvoří správný soubor `kostky.txt`, ale vytvoří správný soubor `retez.txt`, nedostane žádné body.

Funkčnost	1 bod	testovací adresář 1
	1 bod	testovací adresář 2
	1 bod	testovací adresář 3
	1 bod	testovací adresář 4
	1 bod	testovací adresář 5
	1 bod	testovací adresář 6
	1 bod	testovací adresář 7
	1 bod	testovací adresář 8
	1 bod	testovací adresář 9
dokumentace	1 bod	komentáře, přehlednost, výstižné názvy proměnných, ...