
Hodnocení soutěžních úloh

Kategorie žáci

Soutěž v programování – 23. ročník
Krajské kolo 2008/2009
16. až 18. dubna 2009

Čísla

Koeficient 1

Najděte všechna osmimístná čísla C , pro která platí, že z číslic použitých k jejich zápisu lze složit dvě čtyřmístná čísla A a B a platí $A * B = C$.

Na obrazovku vypište počet nalezených čísel a do souboru `vysledek.txt` uložte nalezená čísla tak, že na jednom řádku bude vždy jedno číslo.

Například číslo 10526400 se započte jenom jednou a rovněž ve výsledném souboru bude uvedeno jenom jednou, přestože je několik možností, jak z jeho cifer složit čtyřmístná čísla vyhovující podmínkám v zadání ($10526400 = 2040 * 5160 = 2064 * 5100$).

Příklad 1. Ukázka možného začátku výstupního souboru

```
10025010
10042510
10052010
10052064
```

Hodnocení

| | | |
|-------------|--------|---|
| funkčnost | 4 body | Program vrátí správný počet nalezených čísel (3382) |
| | 2 body | Ve výstupním souboru nejsou duplicity |
| | 1 bod | Výstupní soubor (po setřídění) souhlasí se vzorovým souborem |
| efektivita | 2 body | Program neprochází všechny možnosti, ale nějakým způsobem před testováním vyloučí variantu jako nemožnou. Například $A \geq B$, ciferný součet A, B je roven cifernému součtu C , $A * B > 99999999$ |
| dokumentace | 1 bod | komentáře, přehlednost, výstižné názvy proměnných, ... |

Pro zjištění toho, zda jsou v souboru duplicity, můžete použít následující postup:

```
sort vysledek.txt > setrideno.txt
sort -u vysledek.txt > unikatni.txt
fc setrideno.txt unikatni.txt /N
```

Pokud v souboru s výsledky nebyly duplicity, označí program **fc** soubory za identické.

To, že je výsledek správný, lze ověřit porovnáním výše vygenerovaného souboru `unikatni.txt` se vzorovým výsledkem `vysledek.txt`. Porovnání je opět možné provést pomocí programu **fc**.

Postřehoměr

Koeficient 2

Vaším úkolem je napsat program, který zjistí, jak rychle dokáže uživatel zareagovat na změnu vybraného prvku na obrazovce. Program po svém spuštění vyzve uživatele k odstartování pokusu. Když uživatel pokus odstartuje, počká program náhodně zvolený čas v rozsahu 1.5–10 sekund. Po uplynutí tohoto času se uživateli zobrazí výzva (například nápis TEĎ) a začne se měřit čas. Ve chvíli, kdy uživatel zareaguje (stiskne tlačítko či klávesu), měření času se ukončí a zobrazí se čas uplynulý od zobrazení výzvy po stisk tlačítka. Následně program umožní provedení dalšího měřeného pokusu. Během řešení úlohy se pokuste zvolit co nejpřesnější metodu měření času.

Hodnocení

| | | |
|-------------|--------|--|
| funkčnost | 4 body | program jde spustit, počká na odstartování měření, ukončení měření, zobrazí výsledný čas |
| | 2 body | program umožňuje opakované měření času |
| efektivita | 1 bod | čas je zobrazen včetně jednotek (např. sekundy, milisekundy) |
| | 1 bod | (odhadovaná) přesnost měření je alespoň 100ms |
| | 1 bod | ovládání programu je intuitivní |
| dokumentace | 1 bod | komentáře, přehlednost, výstižné názvy proměnných, ... |

Hledání palindromů

Koeficient 2

Palindrom je text, který je stejný (až na mezery), když ho čteme zepředu i zezadu.

Příklad: „krk“ je palindrom, „jelenovi pivo nelej“ je palindrom, „abcabc“ není palindrom.

Napište program, který v zadaném textu najde nejdelší palindrom a vypíše jej.

Program čte text z uživatelem zadaného souboru a najdelší palindrom vypíše na obrazovku a zároveň uloží do souboru `vysledek.txt` v aktuálním adresáři.

Pokud je v textu více palindromů maximální délky, stačí vypsát jeden z nich. Zadaný text neobsahuje písmena s diakritikou (háčky a čárky). Při hledání palindromů ignorujte všechny znaky kromě znaků anglické abecedy a–z, A–Z. Velká a malá písmena považujte za ekvivalentní. Písmeno „ch“ pro zjednodušení považujte za dva samostatné znaky.

Příklad 2. Ukázku vstupu a výstupu

Vstup: "Sejdeme se tak jak vloni, u dubu tu budu v osm."

Výstup: "u dubu tu budu"

Vstup: "Nastav ucho hochu a poslechni dobrou radu".

Výstup: "o ho" (*písmeno „ch“ se chápe jako dva znaky, takže nejdelší palindrom není „ucho hochu“*)

Hodnocení

Výstupu programu se nemusí 100% shodovat, mohou se lišit například mezerami a dalšími znaky, které podle zadání nemohou být součástí palindromu.

| | | |
|-------------|--------|--|
| Funkčnost | 1 bod | Pro vstup <code>vstup1.txt</code> vrátí <code>vystup1.txt</code> |
| | 1 bod | Pro vstup <code>vstup2.txt</code> vrátí <code>vystup2.txt</code> |
| | 1 bod | Pro vstup <code>vstup3.txt</code> vrátí <code>vystup3.txt</code> |
| | 1 bod | Pro vstup <code>vstup4.txt</code> vrátí <code>vystup4.txt</code> |
| | 2 body | Pro vstup <code>vstup5.txt</code> vrátí <code>vystup5.txt</code> |
| | 3 body | Pro vstup <code>vstup6.txt</code> vrátí <code>vystup6.txt</code> |
| dokumentace | 1 bod | komentáře, přehlednost, výstižné názvy proměnných, ... |

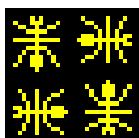
Okrasné rámečky

Koeficient 2

Napište program na přidávání okrasných rámečků do obrázku.

Program načte dva soubory podle volby uživatele: první soubor obsahuje obrázek s rámečky. Druhý soubor je obrázek, do kterého se rámečky umístí. Výsledný obrázek s rámečky program zobrazí a případně uloží na místo podle volby uživatele. První soubor se použije tak, že se obrázek rozdělí na 4 části tak, že se rozpůlí svisle i vodorovně. Tyto čtyři části se umístí do rohů obrázku z druhého souboru.

Obrázek 1. Ukázka prvního souboru s rámečkem



Obrázek 2. Ukázka druhého souboru s obrázkem



Obrázek 3. Ukázka výstupního obrázku po doplnění rámečků



Vstupní obrázky mohou být v libovolném z formátů BMP, GIF, JPEG a PNG. Výstupní obrázek musí být uložen v jednom z formátů BMP, GIF, JPEG nebo PNG – nejlépe ve stejném formátu jako měl druhý vstupní soubor.

Při psaní programu můžete předpokládat, že maximální velikost prvního souboru s rámečky je 400 × 300 pixelů a druhého souboru s obrázkem pak 1024 × 768 pixelů. Velikost výstupního souboru je vždy stejná jako druhého vstupního souboru.

Program můžete dále rozšířit o následující funkce:

- Animaci rozdělení rámečků obrázku prvního souboru při současném zobrazení nad obrázkem druhého souboru.
- Spojení rohů rámečku opakováním jednoho nebo dvou prostředních pixelů.

Obrázek 4. Ukázka spojených rámečků



- Pro formáty BMP a JPEG možnost zadání barvy, která se má v obrázku rámečku chovat jako průhledná. U formátů GIF a PNG využít informaci o průhlednosti uloženou v přímo v grafickém souboru.
- Možnost doplnění stejného rámečku do více vybraných obrázků najednou.

Hodnocení

| | | |
|-------------|----------|---|
| funkčnost | 0.5 bodu | první a druhý soubor lze vybrat interaktivně |
| | 1 bod | program správně zobrazí rámečky (1. soubor 192x128.bmp) nad obrázkem (2. soubor Duncan400x300.jpg) |
| | 1 bod | program správně zobrazí rámečky i pro největší soubor (1. soubor 400x300.bmp) nad obrázkem (2. soubor Duncan1024x768.jpg) |
| | 0.5 bodu | uživatel si může vybrat výstupní soubor |
| | 1 bod | vstupní soubor je správně uložen |
| | 1 bod | Za každý podporovaný formát na vstupu 0.25 bodu |
| | 1 bod | program ukazuje animaci rozdělení |
| | 1 bod | program zvládne spojení rohů |
| | 0.5 bodu | program zvládá průhlednost ručním výběrem průhledné barvy |
| | 0.5 bodu | program zvládá průhlednost uloženou v grafickém souboru (soubory vselogo.gif nebo a-alfa.png) |
| | 1 bod | program umí více obrázků najednou |
| dokumentace | 1 bod | komentáře, přehlednost, výstižné názvy proměnných, ... |

Rychle pryč

Koeficient 3

Váš kamarád Tomáš Marný vás pozval, abyste si s ním zahráli jednu **podoomacku** připravenou hru. Když jste nic zlého nečekaje k Tomášovi dorazili, zjistili jste, že přestavěl svůj pokoj na bludiště a že v něm jste, světe div se, zatím sami. Nicméně by bylo velmi rozumné se z bludiště co nejrychleji dostat.

Napište program, který na vstupu dostane soubor s popisem bludiště. Bludiště se skládá z S krát V políček, každé políčko může být prázdné nebo to může být stěna. Vaším úkolem je najít nejkratší cestu ven z bludiště, pokud nějaká existuje, přičemž pohybovat se můžete doleva, doprava, nahoru, dolů (tj. na západ, východ, sever a jih) a cesta samozřejmě smí vést jenom po prázdných políčkách.

Popis vstupu

Vstup načítejte ze souboru. Na prvním řádku jsou mezerou oddělená dvě čísla S a V , $1 \leq S \leq 200$, $1 \leq V \leq 200$, která určují šířku a výšku bludiště. Na následujících V řádcích se nachází popis bludiště samotného, každý řádek obsahuje přesně S znaků s následujícím významem:

| Znak | Políčko |
|------|-----------|
| . | volné |
| X | zeď |
| @ | vy |
| § | cesta ven |

Bludiště obsahuje přesně jeden znak @ a jeden znak §.

Popis výstupu

Výstup vypíšete do souboru. Pokud z bludiště neexistuje cesta ven, vypíšete do souboru jedinou řádku "Mas to marny.". V opačném případě musí první řádek výstupního souboru obsahovat délku nejkratší cesty mezi políčky @ a §. Dalších V řádků obsahuje mapu bludiště, na které je nejkratší cesta vyznačená – všechna volná políčka na nejkratší cestě vyznačte znakem *.

Příklad 3. Ukázkové vstupy a výstupy

```
vstup   výstup
-----
7 5     15
@XXXXXX @XXXXXX
.X$. . . . *X$**..
.XXX. . . *XXX**..
.X..X.. *X..X*.
..... *****.
```

```
vstup   výstup
-----
5 3     Mas to marny.
XX..$
..X..
@..XX
```

Hodnocení

Funkčnost se zkouší na devíti vstupních souborech, za každý správně vyřešený dostane soutěžící 1 bod. Za správně vyřešený vstup se považuje takový, že

1. program na tomto vstupu doběhne do pěti vteřin,
2. program vypíše délku cesty, která se shoduje s následující tabulkou,
3. program vypíše cestu, která vede bez přerušení ze startu do cíle (stačí zkontrolovat pohledem).

| | | |
|-------------|-------|--|
| funkčnost | 1 bod | pro vstup z1.in je délka 30 |
| | 1 bod | pro vstup z2.in je délka 66 |
| | 1 bod | pro vstup z3.in je délka 3233 |
| | 1 bod | pro vstup z4.in je délka 136 |
| | 1 bod | pro vstup z5.in je délka 262 |
| | 1 bod | pro vstup z6.in je délka 9724 |
| | 1 bod | pro vstup z7.in je délka 352 |
| | 1 bod | pro vstup z8.in je délka 282 |
| | 1 bod | pro vstup z9.in je délka 20099 |
| dokumentace | 1 bod | komentáře, přehlednost, výstižné názvy proměnných, ... |