
Zadání soutěžních úloh

Kategorie programování žáci a mládež

19. až 21. dubna 2018

Soutěž v programování – 32. ročník

Krajské kolo 2017/2018

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 (vhodné členění zdrojového kódu, vhodně zvolené názvy identifikátorů, komentáře na místech, kde je to potřeba, atd.). Body získané za každou úlohu se ještě násobí koeficientem, který odráží složitost úlohy.

Na řešení úloh máte 4 hodiny čistého času.

Před zahájením soutěže vám pořadatel oznámí, kde najdete testovací soubory a kam máte ukládat řešení úloh.

Hodiny jako teploměr

Koeficient 1

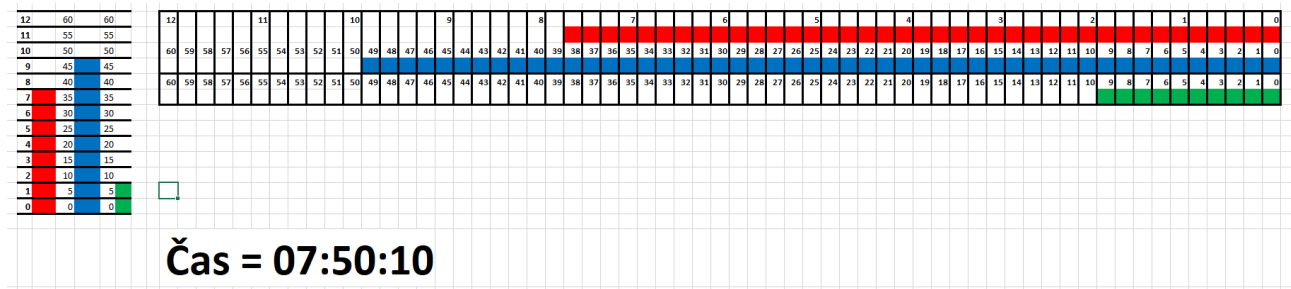
Karlík si takhle jednou hrál na doktora a v ruce měl teploměr, prohlížel si jej a řekl si, že by bylo fajn, kdyby i jeho hodiny na stěně ukazovaly čas stejně, jako se ukazuje teplota na teploměru. Zkuste Karlíkovi pomoci a napište pro něj program, který bude ukazovat aktuální čas na 3 stupnicích (hodiny, minuty a sekundy zvlášť) podobně jako teploměr.

- Hodiny jsou ukázány jako červený sloupec, stupnice je označena od 0 do 12. (12:30, tedy půl jedné odpoledne, se na hodinách zobrazí stejně jako 0:30, tedy půl jedné v noci)
- Minuty jsou ukázány jako modrý sloupec, stupnice je označena od 0 do 60.
- Sekundy jsou ukázány jako zelený sloupec, stupnice je označena od 0 do 60.
- Všechny stupnice musí být čitelné, minimální velikost hodin je 400 x 400 bodů (pixelů).

Jako vylepšení můžete naprogramovat (a získat tak další body):

- Plynulý pohyb sloupce hodin a minut (rostou s každou sekundou, minutou).
- Možnost nastavení času na hodinách (jen na hodinách, ne v celém počítači).

Jak by mohly takové hodiny vypadat se podívejte na obrázku, berte to jako inspiraci :-)



Čtení morzeovky s výpisem na obrazovku

Koeficient 1

Napište program který umí luštit zprávy psané v morzeově abecedě. Vstupní text obsahuje malá a velká písmena české abecedy, mezery, konce řádků a interpunkci a je kódovaný následným způsobem:

- Pokud slovo začíná samohláskou, interpretuje se toto slovo jako tečka morzeovy abecedy.
- Pokud slovo začíná souhláskou, interpretuje se jako čárka morzeovy abecedy.
- Interpunkční znaménka oddělují jednotlivá písmena.

Příklady

Vstup:

```
Aby program; óóóóóhromil porotu tak musí;
umět;všechny,soubory jedním spuštěním.
Musí mít READ ,
ME alternativně help!
```

Výsledek: AJET00K

Vstup:

```
ó n, á h l ý !d é š? ě j i ž ,z v í ,ř i l ,p r a c, h a č, i l
,á; l a, ň t e ,ď b ě ,ž í s h, o u f ,c e m ,G a z e, l k ,ú k,
r y t, ů m;
```

Výsledek: APKQGKQKAENGGYUKCMAKA

Od programu se očekává, že po startu umožní opakovaně vybírat vstupní soubory a dešifrovaný obsah vypisovat na obrazovku. Tj. na jedno spuštění zpracovat postupně více vstupních souborů.

Předpokládejte, že soubory jsou uloženy v kódování UTF-8.

Morzeova abeceda

A	.-	H	N	-.	U	...-
B	-...	CH	----	O	---	V	...-
C	-.-.	I	..	P	W	..--
D	-..	J	Q	---.	X	...-
E	.	K	.-.	R	...	Y	-.--
F	L	S	...	Z	---.
G	--.	M	--	T	-		

Počítání skvrn

Koeficient 1

V akademii věd v chemické sekci manuálně počítají olejové skvrny s různou barvou na vodním filmu. Pomozte jim a napište program, který jim tu práci usnadní.

Pro zjednodušení předpokládejte:

- že, stačí analyzovat pouze statický obraz v souboru PNG (maximální velikost 2000x2000 pixelů),
- že, okolní barva filmu je bílá,
- že, skvrny mají kruhový tvar a minimální velikost je 1x1 pixel,
- že, jedna skvrna má pouze jednu barvu,
- že, žádná skvrna nemá bílou barvu,
- že, může být více skvrn s téže barvy,
- že, skvrny nezasahují mimo analyzovaný prostor a ani se nedotýkají stěn,
- že, se skvrny nepřekrývají a ani se nedotýkají (mají mezi sebou minimálně 1 pixel vodního filmu a to i v šikmém směru).

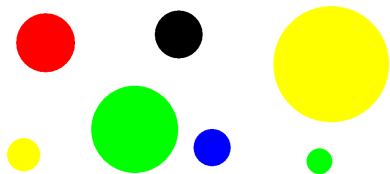
Dále zajistěte:

- výběr souboru PNG pomocí dialogového okna,
- výběr souboru lze provést opakovaně bez ukončení programu,
- program si pamatuje poslední použitý adresář.


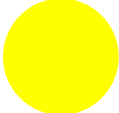
Program musí:

1. Spočítat a vypsát celkový počet skvrn.
2. Spočítat a vypsát počet skvrn od každé barvy – barvy vyjádřete ve formátu [red,green,blue] (řazeno od nejpočetnější barvy po nejméně početnou barvu, dále řazeno dle formátu [r,g,b]).
3. Seřadit skvrny podle velikosti a zobrazit je jako obrázek. Skvrny uspořádejte do původní šířky obrázku ve vodorovných řadách od shora s distancí 5 pixelů tak, aby nezasahovaly do okraje obrázků. Pokud výška obrázku nestačí zvětšete ji.

Ukázka zadání: Vodní film se skvrnami – vstupní soubor:



Ukázka řešení:

```
úkol 1: celkový počet skvrn: 7
úkol 2: [0,255,0]: 2x, [255,255,0]: 2x,
        [0,0,0]: 1x, [0,0,255]:1x, [255,0,0]: 1x
        
úkol 3: 
```

V adresáři skvrny/ nejdete soubory pro testování vaší úlohy a ukázkové řešení úlohy.

Úřad

Koeficient 3

Na úřadě v očíslovaných kancelářích sedí úředníci. Pokud jim donesete všechna razítka, která chtějí od jiných úředníků, dají vám to svoje (samozřejmě až po vystání fronty, vyplnění formulářů a zaplacení poplatků, ale to tu řešit nebudeme).

Napište program, který najde nejrychlejší cestu mezi jednotlivými úředníky tak, abyste všechna potřebná razítka ve správném pořadí dostali.

Vstup

Úředníci jsou popsáni přiřazenými čísly, která mají na svých dveřích (v každé kanceláři je jen jeden úředník). Stejným číslem označíme i razítko, které úředník vydává. Kanceláře (tím pádem i úředníci a razítka) mají přiřazená čísla v rozsahu 1..10000. Vstupem je soubor v následujícím tvaru:

- na prvním řádku jsou čísla razítek, která potřebujeme, oddělená mezerou
- další řádky vypadají takhle:

```
<číslo dveří> <požadované razítko 1> <požadované razítko 2> ...
```

- pokud by nějaké číslo dveří v souboru nebylo, předpokládejte, že příslušný úředník výjimečně žádné razítko nevyžaduje

Výstup

Pokud řešení neexistuje z důvodu že si úředníci, jejichž razítka potřebuji, navzájem vyžadují razítka do kruhu, program tento kruh vypíše v následujícím formátu (místo `<u1>` až `<un>` uvedete příslušná čísla):

```
Nelze vyresit - pozadavky do kruhu:
```

```
<u1> chce <u2>
```

```
<u2> chce <u3>
```

```
...
```

```
<un> chce <u1>
```

V ostatních případech bude výstupem seznam dveří, která je potřeba postupně projít, tak abych u každého úředníka už měl všechna razítka, která požaduje, a na konci měl všechna razítka, která potřebuji. Čísla vypisujte jednotlivě po řádcích (na každý řádek jedno číslo).

Ukázky

Vstup:

```
1 2 3
1 2 3 4 5
2 3 4 5
3 4 5
4 5
```

Výstup:

```
5
4
3
2
1
```

Vstup:

```
1 2 3
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5 3
```

Výstup:

```
Nelze vyřešit - požadavky do kruhu:
3 chce 4
4 chce 5
5 chce 3
```