

---

# Zadání soutěžních úloh

## Kategorie mládež

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

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.

## 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
```

## 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.

# 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 nejdelší 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“*)

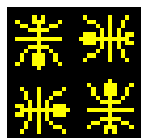
# Okrasné rámečky

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  $400 \times 300$  pixelů a druhého souboru s obrázkem pak  $1024 \times 768$  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.

## 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, může to být stěna nebo to mohou být dveře. Dveře mají vždy nějakou barvu, buď červenou, zelenou, modrou nebo žlutou. Na prázdných políčkách se mohou nacházet klíče, každý má také nějakou barvu, červenou, zelenou, modrou nebo žlutou. V bludišti se můžete pohybovat po prázdných políčkách, doleva, doprava, nahoru nebo dolů (tj. na západ, východ, sever a jih). Pokud stoupnete na políčko s klíčem, seberete ho. Na dveře nějaké barvy můžete stoupnout jenom tehdy, když máte klíč příslušné barvy. Jeden klíč můžete použít na otevření libovolně mnoha dveří stejné barvy. Na začátku nemáte přirozeně klíč žádný. Vaším úkolem je najít nejkratší cestu z bludiště splňující popsána pravidla, nebo říct, že žádná neexistuje.

## 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
R, G, B, Y	dveře červené, zelené, modré, žluté
r, g, b, y	klíč červený, zelený, modrý, žlutý

Bludiště obsahuje přesně jeden znak @ a jeden znak §.

## Popis výstupu

Výstup vypište do souboru. Pokud z bludiště neexistuje cesta ven, vypište 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
```

```
vstup      výstup
-----
16 5      39
@X.g...X...yX.. @X.g***X...yX..
.XXXXXX.X.XXXXXX *XXXXXX*X.XXXXXX
...y.rX.G.r..RY$ ***y.rX*G*r**RY$
.XXXXXX.X....XXX *XXXXXX*X....XXX
.....XgbbgX.. *****XgbbgX..
```

```
vstup      výstup
-----
5 3      Mas to marny.
@X.X.
.R.B$
bXrX.
```