

---

# Zadání soutěžních úloh

## Kategorie žáci a mládež

Soutěž v programování – 28. ročník  
Krajské kolo 2013/2014  
10. až 12. dubna 2014

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 úloh máte 4 hodiny čistého času.

Před zahájením soutěže vám pořadatel oznámí, kde najdete testovací soubory.

## Kolo štěstí

Koeficient 2

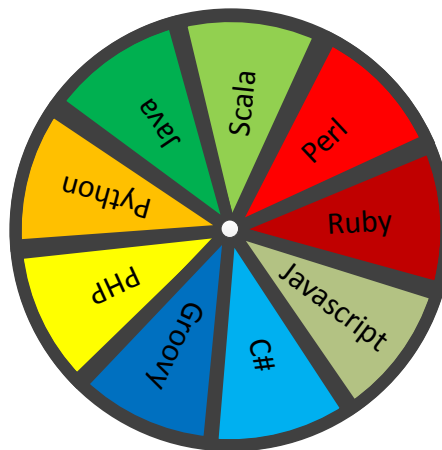
Naprogramujte aplikaci simulující „Kolo štěstí“. Program dovolí do vstupního pole zadat několik řádek textu. Text na jednotlivých řádkách pak zobrazí v kruhových výsečích rozdílných barev. Následně program umožní uživateli kolo „roztočit“ – kolo se pak zastaví na náhodně zvolené výseči a označí ji. Kolo půjde roztáčet opakovaně. Snažte se, aby pohyb kola vypadal realisticky.

### Příklad 1. Příklad vstupu a vykreslení kola štěstí

*Vstup:*

```
JavaScript  
Ruby  
Perl  
Python  
PHP  
Scala  
Java  
C#  
Groovy
```

*Ukázka vykreslení kola štěstí*



# Predjíždějící šneci

Koeficient 1

Vaším úkolem je simulovat závody šneků. Jak takový závod probíhá? Šneci nestartují všichni z jedné startovní čáry, ale jsou umístěni do závodistiště podle své výkonnosti (slabší dostanou náskok), aby byl závod napínavý. Po odstartování se každý šnek začne pohybovat svojí stálou rychlostí směrem vpřed. Závodí se přesně daný čas, po jeho uplynutí se určí pořadí tak, že vyhraje šnek který dolezl nejdále, následuje druhý a tak dále.

A co má tedy dělat váš program? Na vstupu dostane text popisující startovní podmínky závodu – na první řádce je jediné číslo určující délku závodu v sekundách. Každá další řádka přísluší vždy jednomu starujícímu šnekovi a obsahuje dvě čísla oddělená mezerou. První číslo na řádce určuje startovní náskok šneka (vzdálenost v milimetrech od pomyslné startovní čáry směrem do cíle), druhá pak rychlost šneka (mm/s). Startovní číslo je dáno pořadím v souboru, první šnek má startovní číslo 1. Text popisující startovní stav bude programu zadán do textboxu či obdobé grafické komponenty. Pokud pracujete v prostředí, kde odvyklým vstupem je textový standardní vstup, můžete použít i ten.

Vaším úkolem je ze zadných vstupních podmínek (vstupního textu) spočítat pořadí šneků při ukončení závodu a vzdálenost od startovní čáry. Pokud více šneků skončí ve stejné vzdálenosti, bude u nich uvedeno stejné umístění. Dalším úkolem je zobrazit průběh závodu na závodistišti jako animaci, umožněte volbu rychlosti animace.

Pro zjednodušení uvádíme, že v závodu nebude startovat více šneků než 20, maximální trvání závodu je 3600 sekund, pozice a rychlost šneka je celé číslo mezi 0 a 10000. Případné neplatné zadání oznamte uživateli, program samozřejmě nesmí při chybném vstupu havarovat.

## Příklad 2. Příklad vstupu

```
10
350 10
300 20
200 10
250 25
```

## Příklad 3. Příklad výstupu

```
1.-2. místo šneci číslo 2 a 4 (500mm)
3. místo šnek číslo 1 (450mm)
4. místo šnek číslo 3 (300mm)
```

Šnečím závodům nazdar! Zdar!

# Hrátky s čísly

Koeficient 2

*Racionální číslo* je číslo, které lze vyjádřit jako zlomek, tj. podíl dvou celých čísel, většinou zapsaný ve tvaru  $a/b$ , kde  $b$  je různé od nuly.

Desetinný zápis racionálního čísla je periodický. V případě konečného rozvoje – desetinného čísla – tvoří periodu nuly. Pokud se před periodou vyskytuje skupina číslic, která se neopakuje, nazývá se rozvoj *neryze periodický*, neopakující se skupina se nazývá *předperioda*.

Vypracujte program pro určení předperiody a periody podílu  $1/n$  pro maximálně šestimístné přirozené číslo  $n > 1$ . Vstupní hodnoty načtete ze souboru `zadani.txt`, ve kterém je každé číslo na samostatném řádku.

Řádky jsou ukončeny znakem LF.

Program by měl spočítat výsledek pro jednotlivá čísla do jedné minuty.

## Příklad 4. Ukázka výsledku pro vybraná čísla

Vstupní soubor `zadani.txt` obsahuje následujících pět řádek:

```
2
7
192
11111
12345
```

Výsledky spočítané a vypsané programem:

```
1/2=0.5
předperioda: 0.5
perioda: 0

1/7= 0.142857142857142857142857142857142857...
předperioda: 0.
perioda: 142857

1/192 = 0.005208333333333...
předperioda: 0.005208
perioda: 3

1/ 11111= 0.0000900009000
předperioda: 0.
perioda: 00009

1/12345=0.00008100445524503847711624139327663021466180639...
předperioda: 0.0
perioda: 00081004455245038477116241393276630214661806399351964358039692
18307006885378695828270554880518428513568246253543944916970433373835560
95585257189145402997164844066423653300931551235317942486836776022681247
46861077359254759011745646010530579181855002025111381125961927906034831
91575536654515998379910895099230457675172134467395706763872012960712839
20615633859862292426083434588902389631429728635074929121101660591332523
```

```
28878088294856217091940056703118671526933981368975293641150263264479546
37505062778452814904819765087079789388416362899959497772377480761441879
30336168489266909680032401782098015390846496557310652085864722559740785
74321587687322802754151478331308221952207371405427298501417577966788173
34953422438234102875658161198865937626569461320372620494127176994734710
40907249898744430943701903604698258404212231672742
```

## Nejčastější slova strýčka Skrblíka

Koeficient 3

Strýček Skrblík je známý držgrešle, který musí mít vždy všeho nejvíc. Dnes ráno uviděl své synovce, jak si čtou na tabletu knížku, a říkal si: „To je ale slov, ta všechna chci. Ale vlastně chudá slova nechci. Chci jenom slova, která jsou úplně nejbohatší!“ Samozřejmě podle strýčka Skrblíka je nejbohatší to slovo, které se v textu vyskytuje co nejčastěji.

Napište program, který po zadání jména souboru s knihou vypíše tři nejčastější slova, která se v něm nacházejí. Můžete předpokládat, že všechna tato tři slova jsou určena jednoznačně. Váš program by měl fungovat co nejrychleji, jinak Skrblík ztratí trpělivost a moc štědře vás za vaši práci neodmění.

### Popis vstupu

Po spuštění programu umožněte zadání jména souboru s knihou.

Soubor s knihou obsahuje pouze malá a velká písmena anglické abecedy, číslice, mezery a konce řádku, které jsou reprezentovány znakem LF. Slovo je posloupnost písmen a číslic ohraničená mezerou či koncem řádku. Dvě slova jsou stejná, pokud se shodují *bez ohledu na velikost písmen*.

Soubor s knihou může být opravdu veliký – pro dosažení plného počtu bodů by měl váš program fungovat i pro soubor délky 100 MB. Můžete nicméně spoléhat na to, že *jedinečných slov* je v souboru nanejvýš *deset tisíc* a že každé slovo má délku nanejvýš *tisíc* znaků.

### Popis výstupu

Na výstup vypište tři slova, která se v daném souboru vyskytují nejčastěji (bez ohledu na velikost písmen). Slova vypište od nejčastějšího po nejméně časté a u každého vypište počet výskytů. Můžete předpokládat, že tato slova jsou určena jednoznačně.

### Příklad vstupu a výstupu

Soubor s knihou	Výstup programu
Jeden Murphyho zakon o programovani Kazdy program obsahuje jeden chybný radek Libovolny program jde zkratit o jeden radek Z toho plyne ze kazdy jeden program jde zkratit na jeden radek ktery je radek chybný	jeden 5 radek 4 program 3

### Ukázková data

V adresáři `skrblik` se nachází ukázková data včetně příkladu ze zadání. Pro získání plného počtu bodů by měl váš program vyřešit každá z nich do minuty.