

# Kolo štěstí

Koeficient 2

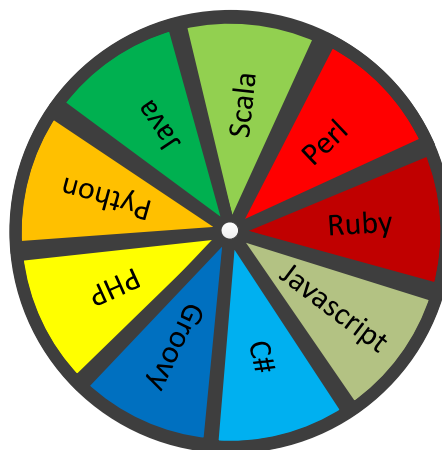
Naprogramujte aplikaci simulující „Kolo štěstí“. Program dovolí do vstupního pole zadat několik řádek textu. Text na jednotlivých řádkách pak zobrazí v kruhových výsečích rozdílných barev. Následně program umožní uživateli kolo „roztočit“ – kolo se pak zastaví na náhodně zvolené výseči a označí ji. Kolo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



## Hodnocení

funkčnost	1 bod	lze zadávat texty pro zobrazení na kole
	1 bod	program správně zobrazí kolo pro dvě řádky vstupu (půl bodu za výseče a půl za texty)
	1 bod	program správně zobrazí kolo pro tři řádky vstupu (půl bodu za výseče a půl za texty)
	1 bod	program správně zobrazí kolo pro čtyři řádky vstupu (půl bodu za výseče a půl za texty)
	1 bod	program správně zobrazí kolo pro text ze zadání (půl bodu za výseče a půl za texty)
	1 bod	barvy výseče se střídají a neopakují
	1 bod	kolo lze roztočit
	0.5 bodu	kolo se po náhodné době zastaví
	0.5 bodu	po zastavení je viditelně označena jedna vybraná výseč
	1 bod	pohyb kola je realistický (rotace zpomaluje před úplným zastavením)
dokumentace	1 bod	dokumentace, komentáře, přehlednost, výstižné názvy proměnných, ...

# 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!

## Hodnocení

funkčnost	1 bod	lze zadat vstupní data způsobem dle zadání (textarea, resp. stdin)
	1 bod	přežije chybné zadání, nezhroutí se, ale oznámí chybu (zkusit soubor chyba1.txt, chyba2.txt - po 0.5 bodu)
	2 body	dobře spočítá pořadí pro vstup1.txt (pořadí: 1. č2 600mm, 2. č4 500mm, 3. č1 450mm, 4. č3 300mm)
	1 bod	dobře spočítá pořadí pro vstup2.txt (pořadí: 1.-2. č2+č4 500mm, 3. č1 450mm, 4. č3 300mm, musí rozpoznat že jde dvou prvních o "první až druhé místo")
	1 bod	dobře spočítá pořadí pro vstup3.txt (pořadí: 1.-3. č1+č2+č4 500mm, 4. č3 300mm, musí rozpoznat že jde o "první až třetí místo")
	1 bod	animace funguje (zobrazuje data bez zjevných chyb)
	1 bod	funguje volba rychlosti animace
	1 bod	animace je přehledná, šneky lze snadno identifikovat, subjektivně graficky hezké řešení
dokumentace	1 bod	dokumentace, komentáře, přehlednost, výstižné názvy proměnných, ...

#### Příklad 4. chyba1.txt

```
50
350 10
a42
250 25
```

#### Příklad 5. chyba2.txt

```
100
350 101
250 102
350 103
250 104
350 105
250 106
350 107
250 108
350 109
250 110
350 111
250 112
350 113
250 114
350 115
250 116
350 117
250 118
350 119
250 120
10350 121
```

### **Příklad 6. vstup1.txt**

```
10
350 10
300 30
200 10
250 25
```

### **Příklad 7. vstup2.txt**

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

### **Příklad 8. vstup3.txt**

```
10
400 10
300 20
200 10
250 25
```

# 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 9. 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 = 0.0052083333333333...
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
```

## Hodnocení

Pokud program včas neskončí (na některém čísle se zasekne) tak 0 bodů za dané číslo.

funkčnost	1 bod	Program načítá data ze vstupního souboru "zadani.txt"
	1 bod	za každý správný 0.25  1/262144= 0.000003814697265625X0 1/128 = 0.0078125X0 1/524288= 0.0000019073486328125X0 1/1000 = 0.001X0
	1 bod	za každý správný 0.25  1/176 = 0.0056X81 1/786432= 0.000001271565755208X3 1/458752= 0.0000021798270089X285714 1/798720= 0.000001252003X205128
	1 bod	za každý správný 0.25  1/111111= 0.X000009 1/7 = 0.X142857 1/56 = 0.017X857142 1/55 = 0.0X18
	1 bod	vše správně 1bod, jinak 0  1/87 = 0.X0114942528735632183908045977 1/870 = 0.0X0114942528735632183908045977 1/8700 = 0.00X0114942528735632183908045977 1/87000 = 0.000X0114942528735632183908045977
	1 bod	vše správně 1bod, jinak 0  1/11 = 0.X09 1/101 = 0.X0099 1/1001 = 0.X000999 1/10001 = 0.X00009999 1/100001= 0.X0000099999
	1 bod	1/917504= 0.00000108991350446X428571
	1 bod	dlouhá perioda, testujeme jen začátek a konec  1/12345 = 0.0X00081004455245038....4212231672742
	1 bod	dlouhá perioda, testujeme jen začátek a konec  1/112265= 0.0X00008907495657595....7758428717766
dokumentace	1 bod	dokumentace, komentáře, přehlednost, výstižné názvy proměnných, ...