1. Športovec prvý deň prebehol x kilometrov. Každý ďalší deň prebehol o 10% viac ako v predchádzajúci deň. Napíš program, ktorý pre dané y zistí, v ktorý deň športovec dokopy prebehne aspoň y kilometrov. Napríklad, po spustení môžeš dostať:

zadaj km pre prvý deň: 10

zadaj cieľové km: 20

na 9. deň prebehne 21.44 km

zadaj km pre prvý deň: 100

zadaj cieľové km: 121

na 3. deň prebehne 121.00 km

1. Budeme konštruovať takúto postupnosť celých čísel:
   * začneme zadaným číslom n
   * ak je párne, vydelíme ho 2
   * inak sa vynásobí 3 a pripočíta 1
   * toto sa opakuje, kým nedostaneme číslo 1

Napíš program, ktorý pre dané štartové číslo vypíše takto skonštruovanú postupnosť. Napríklad:

zadaj číslo: 44

44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

1. Napíš program, ktorý číta desatinné čísla zo vstupu a keď zadáme 0, čítanie čísel zo vstupu skončí a program vypíše súčet všetkých týchto čísel. Použi while-cyklus. Napríklad po spustení:

zadaj 1. číslo: 17

zadaj 2. číslo: 3.14

zadaj 3. číslo: -9.8

zadaj 4. číslo: 6

zadaj 5. číslo: 0

súčet všetkých prečítaných čísel je 16.34

1. Napíš program, ktorý zadané číslo **rozloží na prvočinitele** (vyjadrí ho ako súčin prvočísel). Tento rozklad zapíš v tvare rovnosti s násobením:

zadaj číslo: 60

60 = 2 \* 2 \* 3 \* 5

zadaj číslo: 1001

1001 = 7 \* 11 \* 13

zadaj číslo: 37

37 = 37

V programe bude while-cyklus, v ktorom budeš postupne skúšať deliť zadané číslo rôznymi deliteľmi: ak je číslo týmto deliteľom deliteľné, tak ho vypíšeš a samotné číslo vydelíš týmto deliteľom, inak deliteľ zvýšiš o 1 a celé sa to opakuje.

1. Napíš program, ktorý vypíše cifry zadaného čísla postupným delením desiatimi, teda vo while-cykle vypíšeš poslednú cifru (číslo % 10) a pritom ešte samotné číslo vydelíš 10. Súčasne každú túto cifru pripočítaš do počítadla cs (ciferný súčet). Môžeš dostať takýto výstup:

zadaj číslo: 4132

2

3

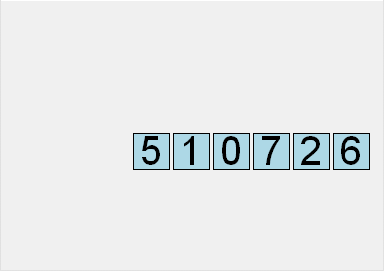
1

4

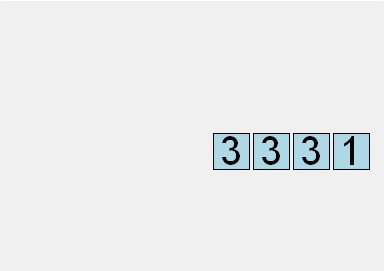
ciferný súčet = 10

Všimni si, že cifry sú vypísané v opačnom poradí ako sú v zadanom čísle.

1. Využi while-cyklus z predchádzajúcej úlohy a vypíš cifry do grafickej plochy (zrejme sprava do ľava). Program jednotlivé cifry vypíše do farebných štvorcov. Napríklad, pre číslo 510726 by si mal dostať:



1. Uprav predchádzajúci program tak, aby sa číslo vypísalo v osmičkovej sústave (zrejme namiesto delenia 10 budeš deliť 8). Pre číslo 1753 by si mal dostať:



Tento výsledok môžeš porovnať s volaním štandardnej funkcie oct (resp. pomocou formátovacej šablóny):

**>>>** oct(1753)

'0o3331'

**>>>** f'*{*1753*:*o*}*'

'3331'

Vyskúšaj upraviť tento program tak, aby vypisoval cifry v dvojkovej sústave. Aj tu môžeš pre kontrolu využiť štandardnú funkciu bin (resp. formátovaciu šablónu):

**>>>** bin(479)

'0b111011111'

**>>>** f'*{*479*:*b*}*'

'111011111'

1. Nasledovný program vykresľuje farebné krúžky v štvorcovej sieti n x n a zafarbuje ich podľa podmienky v príkaze if:

**import** **tkinter**

canvas = tkinter.Canvas()

canvas.pack()

n = 13

**for** i **in** range(n):

**for** j **in** range(n):

x = j \* 20 + 100

y = i \* 20 + 12

**if** i == 5:

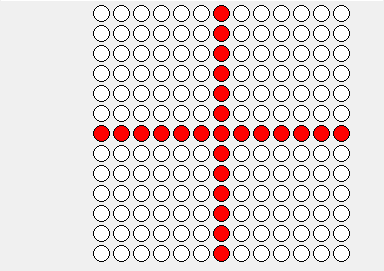
farba = 'red'

**else**:

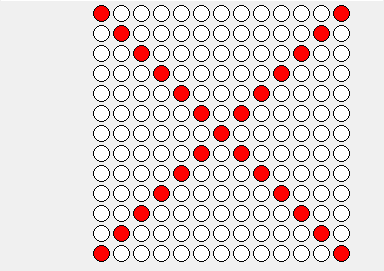
farba = 'white'

canvas.create\_oval(x-8, y-8, x+8, y+8, fill=farba)

**A.** Zmeň túto podmienku tak, aby sa nakreslil obrázok, v ktorom sa zafarbí stredný rad a stredný stĺpec (v programe nemeň iné príkazy, nepridávaj ďalšie):



**B.** Zmeň túto podmienku tak, aby sa nakreslil obrázok, v ktorom sa zafarbia obe uhlopriečky:



Programy by mali fungovať správne aj pri zmenenom rozmere n. Vyskúšaj napríklad n=10.

1. Žiaci sú v rade zoradení podľa veľkosti (od najmenšieho). Napíš program, ktorému najprv postupne oznamujeme (pomocou input) výšky žiakov a ten na záver (zadali sme „prázdnu“ výšku) vypíše, či boli zoradení správne. Použi príkaz while (dopredu nepoznáme počet žiakov), v ktorom bude príkaz if. Mali by fungovať takéto spustenia programu:

zadávaj výšky žiakov

výška 1. žiaka: 100

výška 2. žiaka: 110

výška 3. žiaka: 110

výška 4. žiaka: 120

výška 5. žiaka:

všetci žiaci sú zoradení správne

zadávaj výšky žiakov

výška 1. žiaka: 100

výška 2. žiaka: 120

výška 3. žiaka: 110

výška 4. žiaka: 140

výška 5. žiaka: 145

výška 6. žiaka:

žiaci nie sú správne zoradení

zadávaj výšky žiakov

výška 1. žiaka: 180

výška 2. žiaka:

všetci žiaci sú zoradení správne

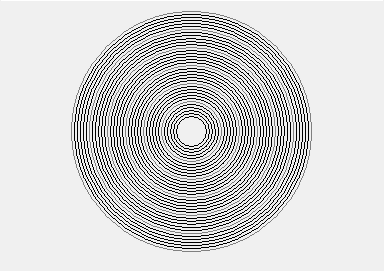
1. Napíš program, ktorý nakreslí gramofónovú LP platňu ako niekoľko sústredných kružníc. Najväčšia z nich má polomer r a každá ďalšia je o 3 menšia. Najmenšia kružnica by nemala mať menší polomer ako 15. Každú k-tu kružnicu nakresli šedou farbou (začni od najväčšej). Napríklad pre premenné:

x, y = 190, 130

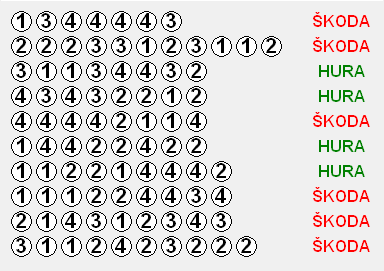
r = 120

k = 6

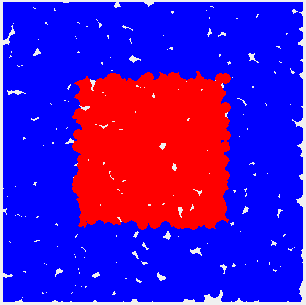
by si mal dostať takýto obrázok:



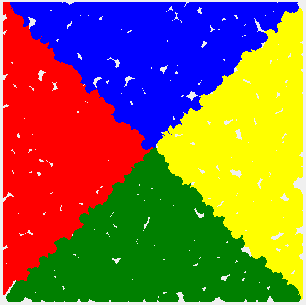
1. Budem hrať takúto hru: kladiem vedľa seba do radu mince s náhodnými hodnotami z <1, 4>; skončím, keď ich súčet bude väčší alebo rovný danému k. Ak skončil so súčtom, ktorý je rovný k, vypíše text 'HURÁ' inak 'ŠKODA'. Napíš program, ktorý túto hru odsimuluje 10-krát a zakreslí to pod seba, napríklad pre ˙˙k=21˙˙ môžeš dostať takýto obrázok:



1. Využi program z prednášky, v ktorom sa vykresľovalo 1000 farebných bodiek (malé krúžky s polomerom 5 bez obrysu) podľa toho či mali x-ovú, alebo y-ovú súradnicu menšiu alebo väčšiu ako 150. Veľkosť grafickej plochy bola 300x300. Zmeň v tomto programe sériu príkazov if tak, aby kreslené bodky vytvorili vnútorný červený štvorec s rozmermi 150x150. Vykresli s 10000 farebnými bodkami:



1. Podobná úloha ako v predchádzajúcom príklade, len v tomto sa využívajú uhlopriečky štvorca. Asi tu budeš vedieť využiť podmienky x < y alebo 300 - x < y:

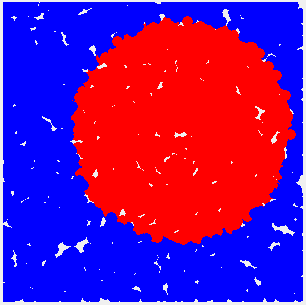


1. V ďalšej verzii bodkovacej úlohy vybodkuješ kruh. Program teraz nakreslí 2000 náhodných bodiek a tie z nich, ktoré majú vzdialenosť od (x0, y0) menšiu ako r zafarbí na červeno, zvyšné na modro. Napríklad pre premenné:

x0, y0 = 180, 130

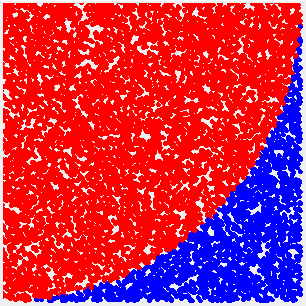
r = 110

by si mal dostať takýto obrázok:



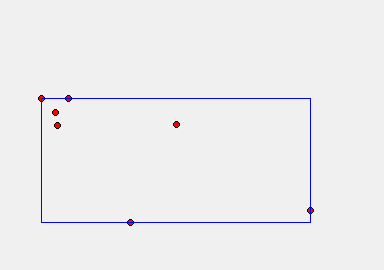
Vzdialenosť dvoch bodov v rovine môžeš počítať podľa vzorca sqrt((x1-x2)\*\*2 + (y1-y2)\*\*2).

1. Aj ďalšia verzia programu generuje náhodné bodky v 300 x 300. Bodiek teraz vygeneruj aspoň 10000 a zmenši ich na polomer 2. Červené bodky budú tie, pre ktoré platí x\*x+y\*y<=300\*300, zvyšné budú modré. Program ma záver vypíše podiel počtu červených ku všetkým bodkám krát 4. Zamysli sa nad tým, prečo sa tento podiel blíži k číslu **pi**. Napríklad:



vyšiel tento podiel 3.1436.

1. Napíš program, ktorý v grafickej ploche najprv vygeneruje n náhodných bodiek (malé kruhy). Potom nakreslí **čo najmenší** obdĺžnik tak, aby sa v ňom nachádzali všetky nakreslené body. Napríklad, pre n=7 môžeš dostať takýto obrázok:

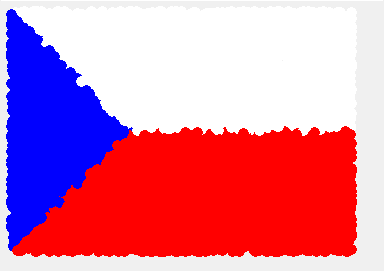


1. Vlajku Nemecka môžeš do štandardnej veľkosti grafickej plochy nakresliť tak, že pomocou cyklu vygeneruješ 10000 náhodných súradníc x z intervalu <10, 350> a y z intervalu <10, 250>. Na vygenerované súradnice nakreslíš farebnú bodku (kruh s polomerom 5 bez obrysu). Farbu bodky zvolíš podľa y-ovej súradnice:
   * ak je y < 90 kresli čierny krúžok,
   * inak, ak je y < 170 kresli čierny krúžok,
   * inak nakresli žltý krúžok.

Mal by si dostať takýto obrázok:



1. Vytvor program, ktorý podobne, ako v predchádzajúcej úlohe, nakreslí vlajku Českej republiky (bývalého Československa):



1. Hrací automat mi pri každom zatlačení (hodil som do neho 1 euro) dá 3 náhodné čísla z <1, 20>. Ak sú medzi týmito tromi číslami dve rovnaké, vyhrávam 5 euro, ak sú všetky tri rovnaké vyhrávam 100 euro. Začal som s nejakou sumou a hral som maximálne 1000 zatlačení, prípadne som skončil skôr, keď som všetko prehral. Napíš program, ktorý to všetko odsimuluje: pri každom zatlačení vypíše '+5' alebo '+100', ak som niečo vyhral, alebo '-1', ak som nič nevyhral. Napríklad:

začínam so sumou: 20

štart -1-1-1-1-1-1-1+5-1-1-1-1+5-1-1-1-1-1-1-1-1+5-1+5-1-1-1-1-1-1+5-1+5

-1-1-1-1-1-1-1-1-1+5+5-1-1-1-1+5-1-1-1+5-1-1-1+5+5-1-1+5+5-1-1+5-1-1-1-1

-1-1-1-1+5-1-1-1+5-1+5-1-1-1-1+5-1+5-1-1+5-1-1-1-1-1-1-1+5-1+5-1-1-1-1-1

+5-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1+5-1-1-1-1-1+5-1

-1-1-1+5-1-1+5-1-1-1-1-1-1-1-1-1-1-1-1

zostalo mi 0 euro

1. Máme mince a bankovky s hodnotami 1, 2, 5, 10, 20, 50, 100. Napíš program, ktorý zistí, ako sa dá poskladať ľubovoľná suma peňazí **minimálnym počtom** kusov peňazí. Použi len jeden for-cyklus, v ktorom bude jeden if, nejaké priradenia a tiež print. Napríklad po spustení:

zadaj cislo: 346

3 krát hodnota 100

2 krát hodnota 20

1 krát hodnota 5

1 krát hodnota 1