1. Napíš funkciu obdlznik(sirka, znak='\*'), ktorá z daného znaku znak vypíše do troch riadkov výstupu obdĺžnik zadanej šírky. Napríklad pre volania:

obdlznik(30, '#')

obdlznik(6)

obdlznik(19, 'O')

dostaneme výstup:

##############################

# #

##############################

\*\*\*\*\*\*

\* \*

\*\*\*\*\*\*

OOOOOOOOOOOOOOOOOOO

O O

OOOOOOOOOOOOOOOOOOO

1. Napíš program na výpočet hodnoty kombinačného čísla n nad k. V programe treba využiť vlastnú funkciu pomocou, ktorej vypočítaš faktoriál.
2. Napíš funkciu riadok(n, text=''), ktorá vypíše n znakový reťazec hviezdičiek '\*', stred ktorého nahradí zadaným textom. Ak je tento zadaný text neprázdny, vloží na jeho začiatok aj koniec medzeru. Napríklad pre volania:

sir = 40

riadok(sir)

riadok(sir, 'Ján Botto')

riadok(sir, 'Žltá ľalija')

riadok(sir, '-')

riadok(sir, 'Stojí stojí mohyla')

riadok(sir, 'Na mohyle zlá chvíľa')

riadok(sir, 'na mohyle tŕnie chrastie')

riadok(sir, 'a v tom tŕní chrastí rastie')

riadok(sir)

dostaneme výstup:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\* Ján Botto \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\* Žltá ľalija \*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* - \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\* Stojí stojí mohyla \*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\* Na mohyle zlá chvíľa \*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\* na mohyle tŕnie chrastie \*\*\*\*\*\*\*

\*\*\*\*\* a v tom tŕní chrastí rastie \*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Napíš funkciu priemer(a, b), ktorá vypočíta priemer dvoch zadaných čísel. Funkcia nič nevypisuje, ale pomocou return vráti vypočítanú hodnotu. Otestuj ju s rôznymi hodnotami parametrov. Napríklad:

**>>>** priemer(1, 4)

2.5

**>>>** priemer(3.14, 31.4)

17.27

1. Na prednáške si sa zoznámil s funkciou fibonacci(n), ktorá počítala n-tý člen **fibonacciho postupnosti**. Napíš funkciu fibonacci\_medzi(od, do), ktorá vypíše (pomocou print) všetky fibonacciho čísla z daného intervalu <od, do>. Táto funkcia by mala obsahovať len jeden while-cyklus (okrem priradení a if). Napríklad pre volania:

fibonacci\_medzi(10, 100)

fibonacci\_medzi(1000, 3000)

dostaneme výstup:

13 21 34 55 89

1. 4
2. Výpis deliteľov zadaného prirodzeného čísla a informácie o tom, či je dané číslo prvočíslom. V programe treba využiť vlastné funkcie s návratovou hodnotou.
3. Napíš funkciu vyhod\_medzery(text), ktorá zo zadaného textu vyhodí všetky medzery. Nepoužívaj žiadne funkcie ani operácie s reťazcami, ktoré sme sa ešte neučili. Funkcia nič nevypisuje, ale pomocou return vráti nový reťazec. Otestuj ju s rôznymi hodnotami parametrov. Napríklad:

**>>>** vyhod\_medzery(' mám rád Python ')

'mámrádPython'

**>>>** vyhod\_medzery(' ')

''

1. Napíš funkciu hadanie(od, do), pomocou ktorej sa budeš vedieť zahrať s počítačom takúto hru: počítač si náhodne pomyslí číslo z intervalu <od, do> (neprezradí nám ho) a my sa ho budeme na maximálne 10 pokusov snažiť uhádnuť. Po každom pokuse nám oznámi, či náš typ je menší ako jeho číslo alebo väčší. Priebeh hry by mohol vyzerať, napríklad takto:

**>>>** hadanie(1, 100)

Myslím si číslo, uhádni ho!

tvoj tip: 50

\*\*\* pridaj

tvoj tip: 75

\*\*\* pridaj

tvoj tip: 88

\*\*\* uber

tvoj tip: 81

\*\*\* uber

tvoj tip: 78

\*\*\* uber

tvoj tip: 77

Uhádol si na 6. pokus. Gratulujem.

**>>>** hadanie(1, 100)

Myslím si číslo, uhádni ho!

tvoj tip: 10

\*\*\* pridaj

tvoj tip: 20

\*\*\* pridaj

tvoj tip: 30

\*\*\* pridaj

tvoj tip: 40

\*\*\* pridaj

tvoj tip: 50

\*\*\* pridaj

tvoj tip: 60

\*\*\* uber

tvoj tip: 59

\*\*\* uber

tvoj tip: 58

\*\*\* uber

tvoj tip: 57

\*\*\* uber

tvoj tip: 56

\*\*\* uber

Neuhádol si ani na 10 pokusov.

Myslel som si číslo 54.

Ak sa budeš hrať so svojim programom, mal by si vždy uhádnuť aj pre interval hadanie(1, 500)

1. Prevod zadaného čísla z desiatkovej sústavy do sústavy so základom, ktorý zadá užívateľ a opačne. Smer prevodu bude možné vybrať v jednoduchom menu. V programe treba využiť vlastné funkcie s návratovou hodnotou.
2. Do daného programu dopíš dve chýbajúce funkcie koleso a doska tak, aby si dostal daný obrázok.

**import** **tkinter**

**def** koleso(...):

...

**def** doska(...):

...

**def** maly\_vozik(x, y):

doska(x, y)

koleso(x-30, y)

koleso(x+30, y)

**def** velky\_vozik(x, y):

doska(x, y, 150, 40, 'green')

koleso(x-35, y, 25, 'orange')

koleso(x+35, y, 25, 'orange')

canvas = tkinter.Canvas()

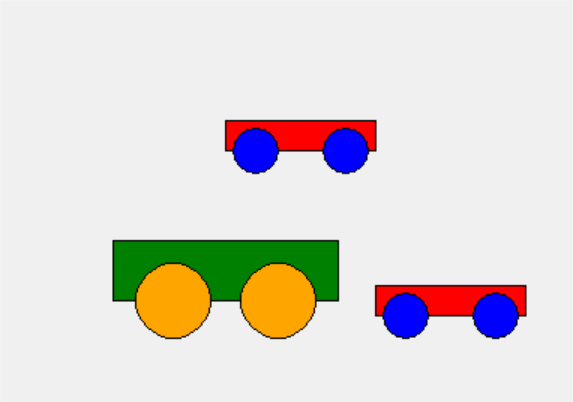
canvas.pack()

maly\_vozik(200, 100)

velky\_vozik(150, 200)

maly\_vozik(300, 210)

tkinter.mainloop()

[](https://python.input.sk/_images/05_c01.png)

1. Napíš funkciu kruhy(x, y), ktorá nakreslí 10 sústredných náhodne zafarbených kruhov, ich polomery budú 5, 10, 15, … Napríklad pre volanie:

**for** i **in** range(10):

kruhy(random.randint(50, 330), random.randint(50, 210))

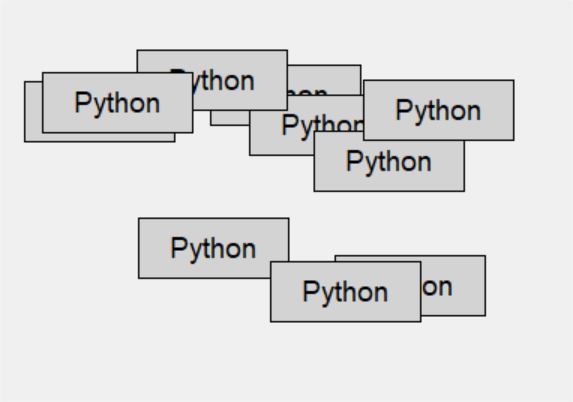
by sa nakreslilo niečo takéto:

[](https://python.input.sk/_images/05_c02.png)

1. Napíš funkciu karticka(x, y, text), ktorá nakreslí bledošedý obdĺžnik a do jeho stredu vypíše zadaný text. Stred kartičky má súradnice (x, y) a jej strany majú dĺžky 100 a 40. Font písma môže byť, napríklad 'arial 14'. Otestuj náhodným vygenerovaním 10 kartičiek, napríklad:

**for** i **in** range(10):

karticka(random.randint(50, 300), random.randint(50, 200), 'Python')

[](https://python.input.sk/_images/05_c03.png)

1. Napíš funkciu dom(x, y, vel1, vel2), ktorá nakreslí domček: štvorec má ľavý dolný roh (x, y) a veľkosť strany je vel1, trojuholník má výšku vel2 a základňu vel1. Oba sú zafarbené rôznymi náhodnými farbami. Napríklad pre volanie:

x, y = 10, 150

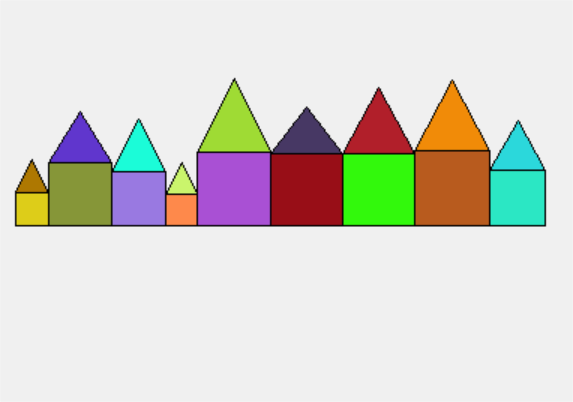
**while** x < 330:

v = random.randint(20, 50)

dom(x, y, v, random.randint(v//2, v))

x += v

by sa nakreslilo:

[](https://python.input.sk/_images/05_c04.png)

1. Napíš funkciu stv(riadok, stlpec, farba='white'), ktorá nakreslí farebný štvorec do myslenej štvorcovej siete, v ktorej je každé políčko veľké 30x30. Ľavý horný roh najľavejšieho horného štvorca má súradnice (5, 5). Napríklad pre volanie:

**for** i **in** range(8):

**for** j **in** range(12):

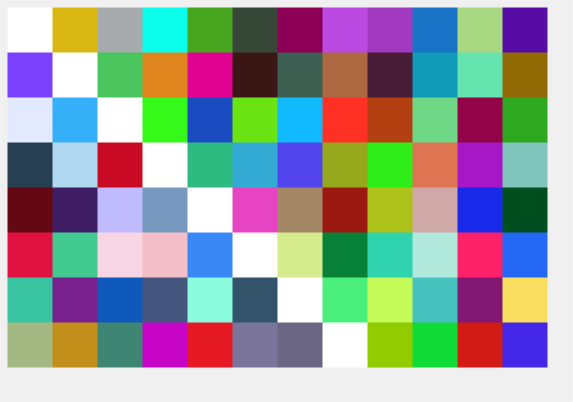
**if** i == j:

stv(i, j)

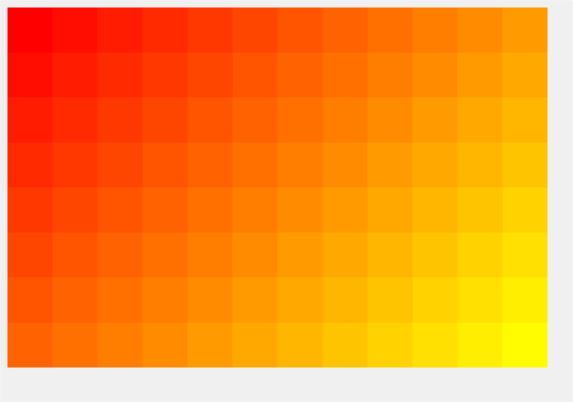
**else**:

stv(i, j, nahodna\_farba())

by sa nakreslilo:

[](https://python.input.sk/_images/05_c06.png)

1. Napíš funkciu rgb (z prednášky) a pomocou nej zafarbi štvorce takto: ľavý horný štvorec má farbu rgb(255, 0, 0) a pravý dolný skoro rgb(255, 255, 0), farba v ostatných štvorcoch plynulo prechádza - čím je štvorec bližšie k pravému dolnému rohu, tým je bližšie k žltej:

[](https://python.input.sk/_images/05_c07.png)

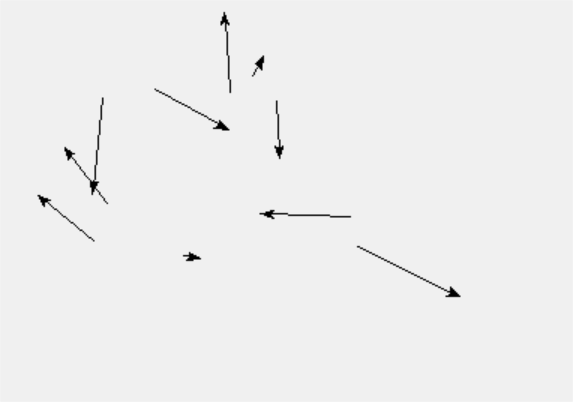
1. Vektor si môžeme predstaviť ako úsečku, ktorá je daná jedným vrcholom (x, y), dĺžkou a uhlom. Uvedom si, že koncové body takéhoto vektora ležia na kružnici s polomerom dĺžka a daným stredom (bodom (x, y)). Úsečku nakreslíme tak, aby mala tvar šípky (do create\_line pridáme pomenovaný parameter arrow='last'). Napíš funkciu vektor(x, y, dlzka, uhol). Otestuj, napríklad takto:

**for** i **in** range(10):

vektor(random.randint(50, 300), random.randint(50, 200),

random.randint(10, 80), random.randint(0, 359))

a môžeš dostať:

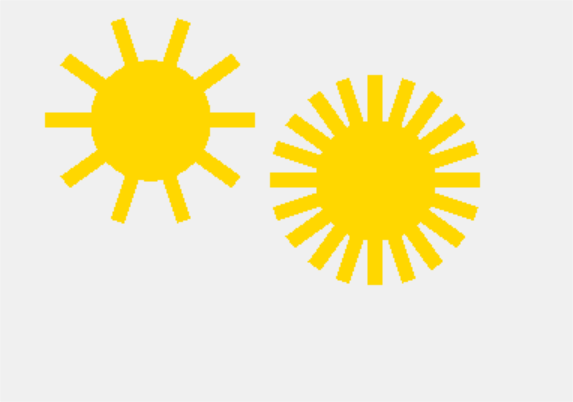
[](https://python.input.sk/_images/05_c08.png)

1. Napíš funkciu slnko(n, x, y), ktorá nakreslí slnko ako n lúčov (hrubšie žlté, resp. zlaté úsečky, ktoré vychádzajú zo stredu (x, y) a majú dĺžku 70) a veľký žltý/zlatý kruh so stredom (x, y) a polomerom 40. Otestuj, napríklad:

slnko(10, 100, 80)

slnko(20, 250, 120)

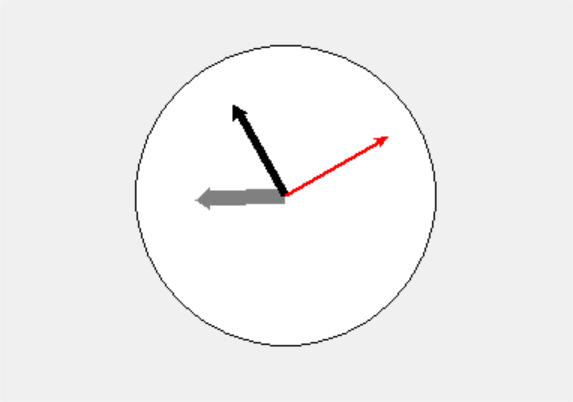
dostaneš:

[](https://python.input.sk/_images/05_c09.png)

1. Napíš funkcie rucicka(uhol, dlzka, hrubka, farba) a hodinky(hod, min, sek), pomocou ktorých nakreslíme ručičkové hodinky. Funkcia rucicka nakreslí len jednu ručičku ako úsečku z bodu (190, 130) pod daným uhlom, danej farby a hrúbky (podobne ako sa kreslil vektor v 15. úlohe). Funkcia hodinky nakreslí ciferník (stačí kruh s polomerom 100) a tri ručičky pre hodiny (dĺžka 60, hrúbka 10, farba 'gray'), pre minúty (dĺžka 70, hrúbka 6, farba 'black'), pre sekundy (dĺžka 80, hrúbka 2, farba 'red'). Napríklad volanie:

hodinky(8, 55, 10)

by nakreslilo:

[](https://python.input.sk/_images/05_c16.png)

Ak by si hodinky zavolal takto:

**import** **time**

**while** **True**:

canvas.delete('all')

h, m, s = time.localtime()[3:6]

hodinky(h, m, s)

canvas.update()

canvas.after(1000)

ukazovali by aktuálny čas.