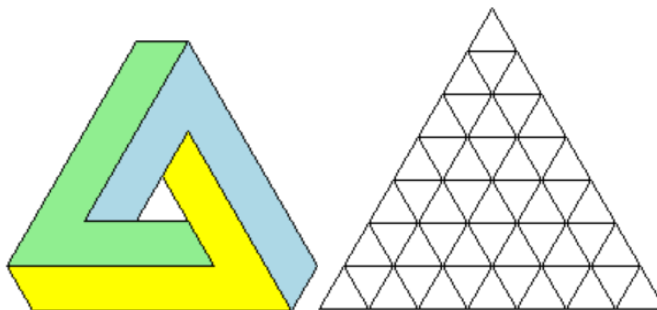


ZBIERKA ÚLOH – ZADANIA

Úloha 1 Poznáte optický klam – Penroseov trojuholník? Zaujímavé je, že si ho v priestore ťažko vieme predstaviť, ale v rovine ho vieme vykresliť. (Zdroj: https://sk.wikipedia.org/wiki/Penroseov_trojuholník)

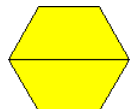
Do zobrazenej trojuholníkovej mriežky zakreslite jeden z farebných dielov uvedeného Penroseovho trojuholníka a popíšte jeho presné rozmery (dĺžky strán a uhly nimi zvierané).



Zakreslite tiež počiatočné pozície vykreslenia všetkých troch dielov Penroseovho trojuholníka.

Úloha 2 Vytvorte programy na vykreslenie uvedených obrázkov, v ktorých použijete vlastnú funkciu pre vykreslenie vzoru.

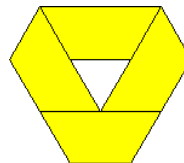
A



B



C



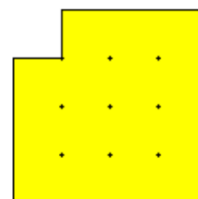
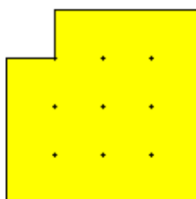
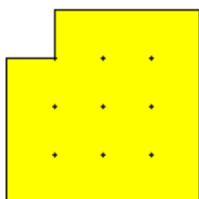
Úloha 3 Na obrázku je vykreslená ozdobná čipka podľa určitého pravidla. V obrázku vyznačte vzory, ktoré sa opakujú.



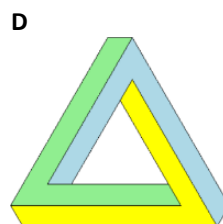
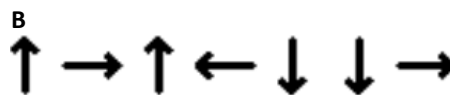
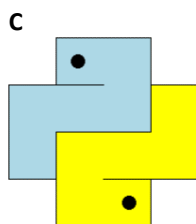
Zapíšte čo najstručnejším spôsobom popis uvedeného obrázka.

.....

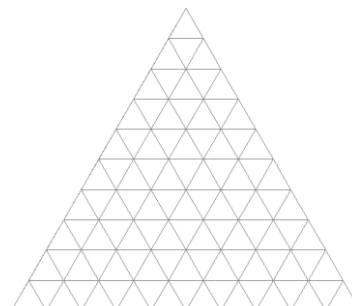
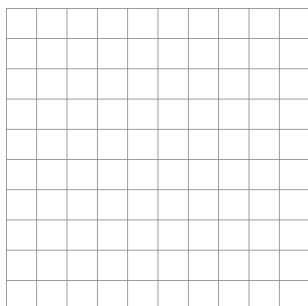
Úloha 4 Máme obrázok s veľkosťou 15 štvorcových jednotiek. Chceme vyrobiť špeciálny puzzle, ktorý obsahuje len rovnaké dieliky. Zakreslite aspoň jeden spôsob pokrytia celého obrázka rovnakými dielikmi.



Úloha 5 Vytvorte programy na vykreslenie uvedených obrázkov.



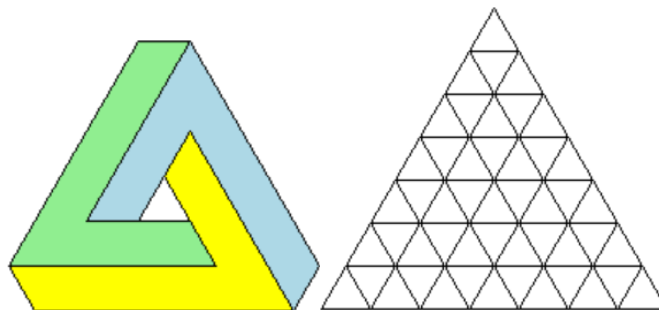
Úloha 6 Navrhните a naprogramujte vlastný zaujímavý obrázok (napr. puzzle, hrací plán stolovej hry, hlavolam, optický klam, logo kapely, čipku či ozdobný ornament), ktorý vznikne opečiatkovaním jedného či dvoch vzorov vo štvorcovej alebo trojuholníkovej mriežke.



ZBIERKA ÚLOH – RIEŠENIA

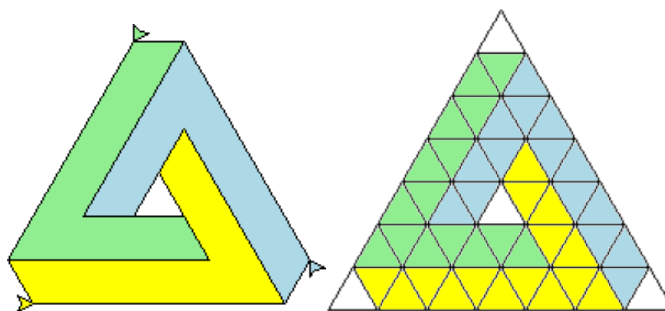
Úloha 7 Poznáte optický klam – Penroseov trojuholník? Zaujímavé je, že si ho v priestore ťažko vieme predstaviť, ale v rovine ho vieme vykresliť. (Zdroj: https://sk.wikipedia.org/wiki/Penroseov_trojuholn%C3%ADk)

Do zobrazenej trojuholníkovej mriežky zakreslite jeden z farebných dielov uvedeného Penroseovho trojuholníka a popíšte jeho presné rozmery (dĺžky strán a uhly nimi zvierané).



Zakreslite tiež počiatočné pozície vykreslenia všetkých troch dielov Penroseovho trojuholníka.

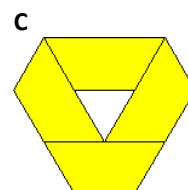
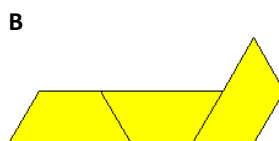
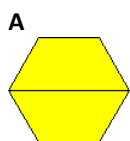
Riešenie:



Dĺžky úsečiek žltého šesťuholníka sú postupne: 5, 4, 1, 2, 4, a 1, uhly natočenia pera vľavo sú postupne 120, 120, 60, -120, 120 a 60 stupňov

Úloha 9 je zameraná na analýzu a presný návrh vykreslenia opakujúceho sa vzoru v uvedenom optickom klame (v Penroseovom trojuholníku). Pri riešení tejto úlohy by si mali žiaci uvedomiť veľký význam náčrtu riešenia a presného návrhu vzoru s uvedenými dĺžkami a uhlami. V zadaní úlohy je uvedený pokyn na vyznačenie vnútorných uhlov úsečiek tvoriacich vzor. Vyznačené vnútorné uhly sú východiskom pre určenie vedľajších (susedných) uhlov, ktoré sa použijú pri samotnom vykreslení vzoru. Súčasťou návrhu vykreslenia obrázku tvoreného vzormi sú tiež počiatočné pozície vykreslenia každého zo vzorom. Aj keď v tejto úlohe žiaci nenapísali ani riadok kódu, je dôležité im zdôrazniť, že vytvorením návrhu vykreslenia urobili podstatnú časť riešenia programátorskej úlohy. Samotné napísanie kódu programu podľa uvedeného návrhu je už len rutinnou prácou vyžadujúcou si len znalosť príkazov korytnačej grafiky.

Úloha 8 Vytvorte programy na vykreslenie uvedených obrázkov, v ktorých použijete vlastnú funkciu pre vykreslenie vzoru.



Úloha 10 nadväzuje na úlohu 2 z predchádzajúcej hodiny vykresľujúcej rovnoramenný lichobežník. Pri jej riešení si žiaci precvičia použitie vlastnej funkcie a príkazov na vypĺňanie uzatvoreného útvaru tvoreného postupnosťou čiar, tvorbu náčrtov s uvedením počiatočných pozícií vykreslenia jednotlivých vzorov v obrázku. Žiaci tiež dostanú príležitosť si uvedomiť výhody opakovaného použitia volania vlastnej funkcie a prehľadnosti uvedeného programového kódu.

Úloha 9 Na obrázku je vykreslená ozdobná čipka podľa určitého pravidla. V obrázku vyznačte vzory, ktoré sa opakujú.

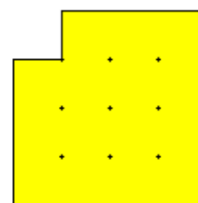
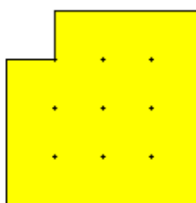
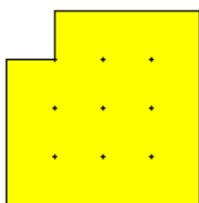


Zapíšte čo najstručnejším spôsobom popis uvedeného obrázka.

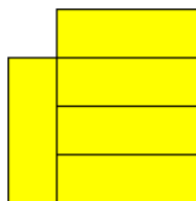
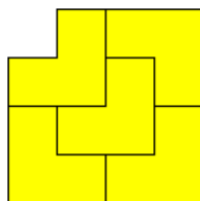
.....

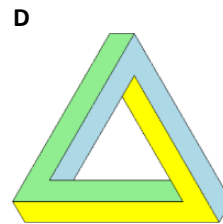
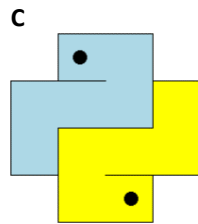
Úloha 11 je zameraná na analýzu obrázka a hľadanie opakujúcich sa vzorov, na stručný zápis postupnosti jednotlivých vzorov. Pri uvedenom zadaní úloha nevyžaduje od žiakov uvedenie žiadneho programového kódu, ani podrobné popisy vykresľovania jednotlivých vzorov.

Úloha 10 Máme obrázok s veľkosťou 15 štvorcových jednotiek. Chceme vyrobiť špeciálny puzzle, ktorý obsahuje len rovnaké dieliky. Zakreslite aspoň jeden spôsob pokrytia celého obrázka rovnakými dielikmi.



Úloha 12 je zameraná na analýzu obrázka – puzzle a návrh aspoň jedného tvaru dielikov, ktorými pokryjeme daný obrázok. Ani táto úloha nepožaduje od žiakov uvedenie programového kódu, ale skôr ich dôsledné matematické uvažovanie a kreativitu. Okrem hraničných riešení – dieliku s 1, resp. 15 štvorcovými jednotkami, očakávame, že žiaci prídu na niektoré z riešení s dielikom s 3 štvorcovými jednotkami (v tvare písmena L či v tvare písmena I).



Úloha 11 Vytvorte programy na vykreslenie uvedených obrázkov.**Riešenie A:**

```
import turtle
```

```
def vecko():
    # kreslenie jednoduchého večka
    pero.right(60)
    pero.forward(-50)
    pero.forward(50)
    pero.right(-120)
    pero.forward(-50)
    pero.forward(50)
    pero.right(60)
```

```
def dvojvecko():
    # kreslenie dvojvečka pozostávajúceho z dvoch večiek
    vecko()
    pero.penup()
    pero.forward(-25)
    pero.pendown()
    vecko()
    pero.penup()
    pero.forward(25)
    pero.pendown()
```

```
def vzor():
    # kreslenie cicmianskeho vzoru pozostávajúceho z dvoch oproti seba vykreslených
    dvojvečiek
    dvojvecko()
    pero.right(180)
    dvojvecko()
    pero.right(180)
```

```
def posun():
    # posunutie pera na nasledovnú pozíciu
    pero.penup()
    pero.forward(100)
    pero.pendown()
```

```
tabula = turtle.Screen()
pero = turtle.Turtle()
pero.pensize(13)
```

```
vzor()
posun()
vzor()
posun()
vzor()
```

```
pero.hideturtle()
tabula.mainloop()
```

Riešenie B:

```
import turtle
```

```
def sipka():  
    # vykreslenie sipky  
    pero.forward(20)  
    pero.left(45)  
    pero.forward(-5)  
    pero.forward(5)  
    pero.right(90)  
    pero.forward(-5)  
    pero.forward(5)  
    pero.left(45)  
    pero.forward(-20)
```

```
def hore():  
    # vykreslenie hornej sipky a posunutie sa na dalsiu poziciu doprava  
    pero.forward(10)  
    pero.left(90)  
    pero.pendown()  
    sipka()  
    pero.penup()  
    pero.right(90)  
    pero.forward(10)  
    pero.forward(5)
```

```
def dole():  
    # vykreslenie dolnej sipky a posunutie sa na dalsiu poziciu doprava  
    pero.forward(10)  
    pero.left(90)  
    pero.forward(20)  
    pero.left(180)  
    pero.pendown()  
    sipka()  
    pero.penup()  
    pero.forward(20)  
    pero.left(90)  
    pero.forward(10)  
    pero.forward(5)
```

```
def doprava():  
    # vykreslenie pravej sipky a posunutie sa na dalsiu poziciu doprava  
    pero.left(90)  
    pero.forward(10)  
    pero.right(90)  
    pero.pendown()  
    sipka()  
    pero.penup()  
    pero.right(90)  
    pero.forward(10)  
    pero.left(90)  
    pero.forward(20)  
    pero.forward(5)
```

```
def dolava():  
    # vykreslenie lavej sipky a posunutie sa na dalsiu poziciu doprava  
    pero.forward(20)  
    pero.left(90)  
    pero.forward(10)  
    pero.left(90)  
    pero.pendown()  
    sipka()  
    pero.penup()  
    pero.left(90)  
    pero.forward(10)  
    pero.left(90)  
    pero.forward(5)
```

```

tabula = turtle.Screen()
pero = turtle.Turtle()
pero.pensize(3)
pero.penup()

```

```

hore()
doprava()
hore()
dolava()
dole()
dole()
doprava()

```

```

pero.hideturtle()
tabula.mainloop()

```

Riešenie C:

```
import turtle
```

```

def hadik():
    # vykreslenie obrysu hadika
    pero.forward(50)
    pero.left(90)
    pero.forward(100)
    pero.left(90)
    pero.forward(100)
    pero.left(90)
    pero.forward(50)
    pero.left(90)
    pero.forward(50)
    pero.left(180)
    pero.forward(100)
    pero.left(90)
    pero.forward(100)
    pero.left(90)
    pero.forward(50)
    pero.left(90)
    pero.forward(50)
    pero.right(90)
    pero.forward(50)
    pero.left(110)
    pero.penup()
    pero.forward(80)
    pero.dot(15, "black")
    pero.forward(-80)
    pero.pendown()
    pero.right(110)

```

```

pero = turtle.Turtle()
tabula = turtle.Screen()

```

```

pero.color("black", "lightblue")
pero.begin_fill()
hadik()
pero.end_fill()

```

```
pero.left(180)
```

```

pero.color("black", "yellow")
pero.begin_fill()
hadik()
pero.end_fill()

```

```

pero.hideturtle()
tabula.mainloop()

```

Riešenie D:

```

import turtle

def diel():
    # vykreslenie dielu s nastavenou farbou vyplne
    pero.begin_fill()
    pero.forward(5 * 30)
    pero.left(120)
    pero.forward((5 - 1) * 30)
    pero.left(120)
    pero.forward(30)
    pero.left(60)
    pero.forward((5 - 3) * 30)
    pero.right(120)
    pero.forward((5 - 1) * 30)
    pero.left(120)
    pero.forward(30)
    pero.left(60)
    pero.end_fill()

def diel_farba1():
    # vykreslenie zlteho dielu
    pero.fillcolor("yellow")
    diel()

def diel_farba2():
    # vykreslenie modreho dielu
    pero.fillcolor("lightblue")
    diel()

def diel_farba3():
    # vykreslenie zeleného dielu
    pero.fillcolor("lightgreen")
    diel()

def trojuholnik():
    # vykreslenie celeho Penroseovho trojuholnika
    diel_farba1()
    pero.forward(5 * 30)
    pero.left(60)
    pero.forward(30)
    pero.left(60)
    diel_farba2()
    pero.forward(5 * 30)
    pero.left(60)
    pero.forward(30)
    pero.left(60)
    diel_farba3()

tabula = turtle.Screen()
pero = turtle.Turtle()

trojuholnik()

pero.hideturtle()
tabula.mainloop()

```

Úloha 13 je zameraná naprogramovanie obrázkov obsahujúcich opakujúci sa vzor. Pri programovaní vykreslenia obrázka A s čičmianskym vzorom (uvedenom tiež v analytickej úlohe 1) môžeme v opakujúcom vzore nájsť vnorené vzory a žiakom uviesť vnorené funkcie (čipka = XXX; X = WW; W = VV). Alternatívou vykreslenia obrázka A je jeho vypísanie do konzoly (>><<>><<>><<).

Pri vykresľovaní obrázku B je vykreslenie vzoru (šípky) jednoduché, náročnejšie je určenie počiatkovej a koncovej pozície vykreslenia každého zo 4 smerov. Zaujímavé na obrázku B je, že znázorňuje návod (na nekorytnačie kreslenie perom, resp. presúvanie sa robota, ktorý nemá nos).

Obrázky C a D vyžadujú na vykreslenie použiť príkazy na farebnú výplň uzatvoreného tvaru. Obrázok C je stredovo symetrickým symbolom jazyka Python s 2 opakovaniami sa vzoru a obrázok D Penroseovho trojuholníka (uvedeného v úlohe 9) s 3 opakovaniami sa vzoru.

Úloha 14 je kreatívna úloha, v ktorej majú žiaci naprogramovať vlastný zaujímavý obrázok obsahujúci jeden či viacero opakujúcich sa vzorov pomocou štvorcovej či trojuholníkovej mriežky. Takýmto obrázkom môže byť, napr. puzzle, hrací plán stolovej hry, hlavolam, optický klam, logo kapely, čipku či ozdobný ornament.