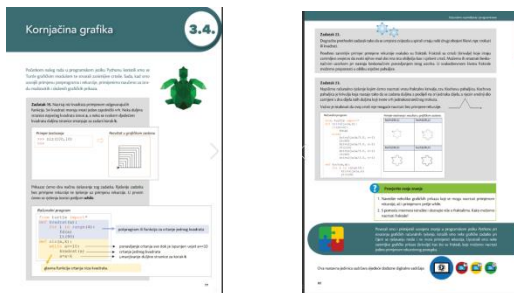


3.4.) Računalno razmišljanje i programiranje: Kornjačina grafika

U bilježnicu napisati podnaslov: „Kornjačina grafika

U udžbeniku pročitati od 77. do 82. stranice, odnosno stranice koje se odnose na navedeni naslov.

<https://www.e-sfera.hr/prelistaj-udzbenik/c0273c34-fb37-4eb1-acba-0a25d493f20f>



Prepisati u bilježnicu (skraćeno, samo bitno):

Fraktali – crteži (krivulje) koje imaju zanimljivo svojstvo da svaki njihov mali dio ima ista obilježja kao i cjeloviti crtež. U svakodnevnom životu možemo ih prepoznati u obliku snježne pahuljice.

Kochova pahuljica - posebna vrsta fraktalnog crteža (krivulje) koji nastaje tako da se zadana duljina a podijeli na tri jednaka dijela, a njezin srednji dio zamijeni s dva dijela istih duljina koji tvore vrh jednakostraničnog trokuta.

Riješiti vježbe i zadatke i rješenja prepisati u bilježnicu:

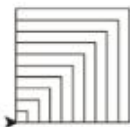
Zadatak 15. Nacrtaj niz kvadrata primjenom odgovarajućih funkcija. Svi kvadrati moraju imati jedan zajednički vrh. Neka duljina stranice najvećeg kvadrata iznosi a , a neka se svakom sljedećem kvadratu duljina stranice smanjuje za zadani korak k .



Primjer testiranja

```
>>> niz(100,10)
>>>
```

Rezultat u grafičkom zaslonu



Prikazat ćemo dva načina rješavanja tog zadatka. Rješenje zadatka bez primjene rekurzije te rješenje uz primjenu rekurzije. U prvom ćemo se rješenju koristiti petljom **while**.

Računalni program

```
from turtle import *
def kvadrat(a):
    for i in range(4):
        fd(a)
        lt(90)
def niz(a,k):
    while a>=10:
        kvadrat(a)
        a=a-k
```

potprogram ili funkcija za crtanje jednog kvadrata

ponavljanje crtanja sve dok je ispunjen uvjet $a \geq 10$

crtanje jednog kvadrata

umanjivanje duljine stranice za korak k

glavna funkcija crtanja niza kvadrata.

Zadatak 16.

Istražimo što će se dogoditi ako zamijenimo redoslijed pozivanja funkcije za crtanje kvadrata `kvadrat(a)` rekurzivne funkcije `niz(a-k, k)`.

```
... else:
    kvadrat(a)
    niz(a-k, k)
```



Vježba 5.

Dogradite računalno rješenje tako da se kvadrati crtaju nekom bojom.



Vježba 6.

Nacrtajte niz šesterokuta koji imaju jedan zajednički vrh. Neka duljina stranice najvećega šesterokuta iznosi **a**. Svakom sljedećem šesterokutu stranica se umanjuje za korak **k**. Stranica šesterokuta ne smije biti manja od deset.

Zadatak 17.

Nacrtajmo spiralu u obliku kvadrata kojoj će duljina najveće stranice biti **a**. Duljinu svake sljedeće stranice treba umanjiti za korak **k** tako da duljina najmanje stranice spirale ne smije biti manja od deset.

Primjer testiranja

```
>>> spirala(100,5)
>>>
```



Rezultat u grafičkom zaslonu



Prikažimo opet dva različita rješenja: rješenje s uporabom rekurzije i bez uporabe rekurzije.

Rješenje primjenom petlje while

```
from turtle import*
def spirala(a, k):
    while a>=10:
        fd(a)
        lt(90)
        a=a-k
```

Rješenje primjenom rekurzije

```
from turtle import*
def spirala(a, k):
    if a<10:
        return
    else:
        spirala(a-k, k)
        fd(a)
        lt(90)
```

Zadatak 18.

Nacrtajmo spiralu trokuta kojoj duljina najveće stranice iznosi **a**. Neka se duljina svake sljedeće stranice umanjuje za korak **k**, a pritom duljina najmanje stranice spirale ne smije biti manja od deset.

Rezultat u grafičkom zaslonu

Primjer testiranja

```
>>> spirala(200,10)
>>>
```



Zadatak 19. Izmijenimo računalno rješenje za crtanje spirale tako da se nakon crtanja svake stranice (na vrhu svake stranice) nacrtaju zvijezda. Neka je **a** duljina najveće stranice spirale. Svaku sljedeću stranicu spirale treba umanjiti za korak **k**. Duljina najmanje stranice ne bi smjela biti kraća od deset.

U primjeru našeg rješenja zvijezda će imati pet krakova. Definirat ćemo listu **boja** koja će sadržavati nazive triju boja ('red', 'blue', 'green'), slučajnim odabirom indeksa (**ind**) program će odabrati jednu od triju boja pri crtanju.

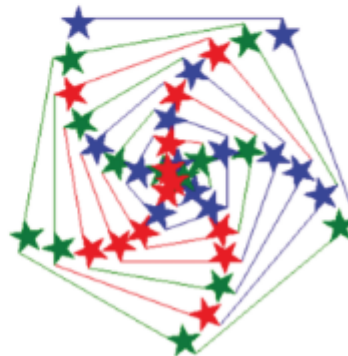
Računalni program

```
from turtle import*
from random import randint
boja = ['red', 'blue', 'green']
def zvijezda():
    ind=randint(0,2)
    color(boja[ind], boja[ind])
    begin_fill()
    for i in range(5):
        fd(30)
        rt(2*360/5)
    end_fill()
def spirala(a,k):
    zvijezda()
    fd(a)
    rt(70)
    if a<10:
        return
    else:
        spirala(a-k,k)
```

Primjer testiranja

```
>>> spirala(200,5)
>>>
```

Rezultat u grafičkom zaslonu

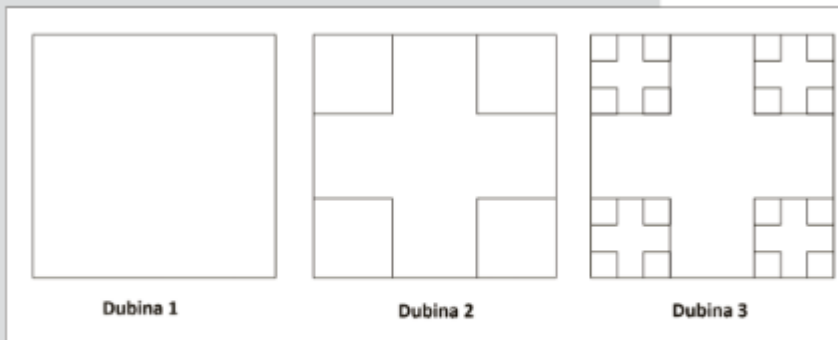


Vježba 7.

Dogradite prethodni zadatak tako da upotrijebite još neke boje osim plave, crvene i zelene.

Zadatak 20.

Definirajmo dubinu kvadrata na sljedeći način. Kvadrat dubine 1 običan je kvadrat stranice **a**. Kvadrat dubine 2 je kvadrat koji u svakom svom vrhu sadržava kvadrat čija je duljina stranice trostruko manja od polazne. Kvadrat dubine 3 je kvadrat čiji kvadrati u svojim vrhovima također sadržavaju kvadrate trostruko kraćih stranica...



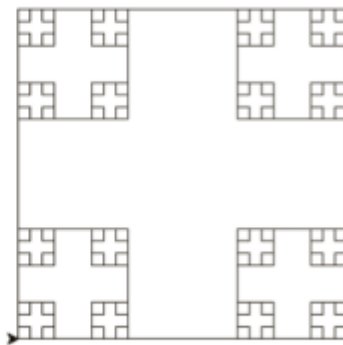
Napišimo računalni program koji će nacrtati kvadrat stranice **a** dubine **n**.

Primjer testiranja

```
>>> kvadrat(300,4)
>>>
```



Rezultat u grafičkom zaslonu



Računalni program

```
from turtle import*
def kvadrat(a,n):
    if n<1:
        return
    else:
        for i in range(4):
            fd(a)
            lt(90)
            kvadrat(a/3,n-1)
```

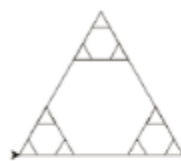
Zadatak 21. Dogradite prethodni zadatak tako da se umjesto kvadrata crtaju trokuti stranice **a** dubine **n**.

Primjer testiranja

```
>>> trokut(200,3)
>>>
```



Rezultat u grafičkom zaslonu



Zadatak 22.

Dogradite prethodni zadatak tako da se umjesto zvijezda u spirali crtaju neki drugi obojeni likovi, npr. trokuti ili kvadrati.

Posebno zanimljiv primjer primjene rekurzije svakako su fraktali. Fraktali su crteži (krivulje) koje imaju zanimljivo svojstvo da svaki njihov mali dio ima ista obilježja kao i cjelovit crtež. Možemo ih smatrati beskonačnim uzorkom jer nastaju beskonačnim ponavljanjem istog uzorka. U svakodnevnom životu fraktale možemo prepoznati u obliku snježne pahuljice.

Zadatak 23.

Napišimo računalno rješenje kojim ćemo nacrtati vrstu fraktalne krivulje, tzv. Kochovu pahuljicu. Kochova pahuljica je krivulja koja nastaje tako da se zadana duljina a podijeli na tri jednaka dijela, a njezin srednji dio zamijeni s dva dijela istih duljina koji tvore vrh jednakostraničnog trokuta.

Važno je istaknuti da ovaj crtež nije moguće nacrtati bez primjene rekurzije.



Računalni program

```
from turtle import*
def krivulja(a,n):
    if n==0:
        fd(a)
    else:
        krivulja(a/3.0, n-1)
        lt(60)
        krivulja(a/3.0, n-1)
        rt(120)
        krivulja(a/3.0, n-1)
        lt(60)
        krivulja(a/3.0, n-1)

def koch(a,n):
    for i in range(3):
        krivulja(a,n)
        rt(120)
```

Primjer testiranja i rezultat u grafičkom zaslonu

koch(100,1)	koch(100,2)
koch(100,3)	koch(100,4)



Provjerite svoje znanje

1. Navedite nekoliko grafičkih prikaza koji se mogu nacrtati primjenom rekurzije, ali i primjenom petlje while.
2. S pomoću interneta istražite i doznajte više o fraktalima. Kako možemo nacrtati fraktale?

Riješiti radnu bilježnicu zadatke od 1. do 5. na stranici 26 i 27 (ponekad su to različiti brojevi stranica zbog razlike u izdanju).

Poslati poruku o izvršenom zadatku putem Teams-a u tim Informatika8.

3.4. Kornjačina grafika

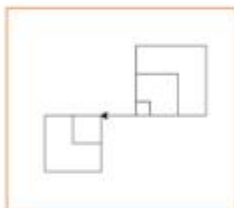
1. Promotri i analiziraj sljedeće računalno rješenje.

Poziv funkcije (primjer testiranja)

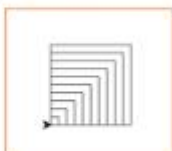
```
>>> niz(100, 20)
```

Koji od navedenih crteža predstavlja rezultat računalnog rješenja?

a)



b)



c)



Računalno rješenje

```
from turtle import *  
def kvadrat(a):  
    for i in range(4):  
        fd(a)  
        lt(90)  
        fd(50)  
def niz(a, k):  
    if a < 10:  
        return  
    else:  
        niz(a-k, k)  
        kvadrat(a)
```

2. Izmijeni računalni program iz prethodnog zadatka tako da program crta niz jednakostraničnih trokuta s duljinom stranice **a**. Duljina svakoga sljedećeg trokuta umanjuje se za korak **k**.

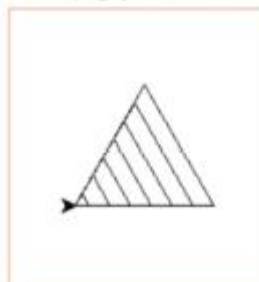
Računalni program



Primjer testiranja

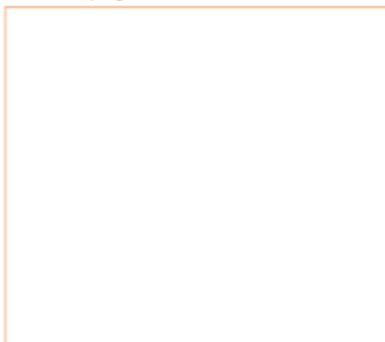
```
>>> niz(100, 15)
```

Rezultat u grafičkom zaslону



3. Nacrtaj spiralu u obliku kvadrata kojoj će duljina najveće stranice biti x . Duljina je svake sljedeće stranice dvostruko manja, no duljina najmanje stranice spirale ne smije biti manja od deset.

Računalni program



Primjer testiranja

```
>>> spirala(100, 5)
```

Rezultat u grafičkom zaslonu



4. Slika prikazuje računalno rješenje kojim bi trebalo nacrtati niz kvadrata tako da svi kvadrati imaju zajednički lijevi donji vrh. Najveća stranica nacrtanih kvadrata definirana je vrijednošću a . Duljina je svake sljedeće stranice kvadrata upola manja, no duljina najmanje stranice kvadrata ne smije biti manja od deset. Analiziraj prikazano računalno rješenje uz očekivani rezultat izvršenja te ga dopuni odgovarajućim naredbama.

Računalni program

```
from turtle import *
def kvadrat(a):
    for i in range(4):
        _____
        lt(90)
def niz(a):
    if _____:
        return
    else:
        kvadrat(a)
        niz(_____)
```

Primjer testiranja

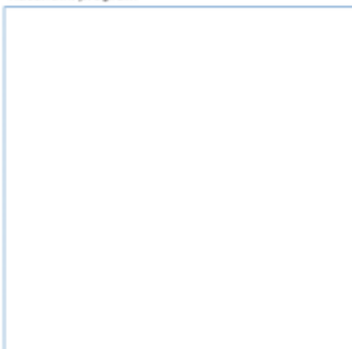
```
>>> niz(100)
```

Rezultat u grafičkom zaslonu



5. Nadogradnjom i izmjenom rješenja iz prethodnog zadatka napravi računalni program koji će nacrtati niz šesterokuta tako da svi šesterokuti imaju zajednički lijevi vrh osnovne stranice. Najveća stranica nacrtanih šesterokuta definirana je vrijednošću a . Duljina je svake sljedeće stranice šesterokuta upola manja, no duljina najmanje stranice šesterokuta ne smije biti manja od deset. Pri izradi rješenja koristi se primjerom testiranja te odgovarajućim rezultatom izvršenja programa.

Računalni program



Primjer testiranja

```
>>> niz(150)
```

Rezultat u grafičkom zaslonu

