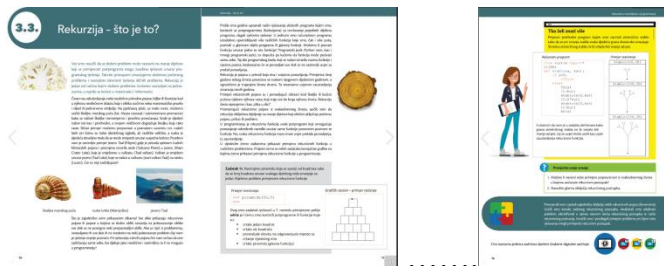


3.3.) Računalno razmišljanje i programiranje: Rekurzija – što je to?

U bilježnicu napisati podnaslov: „Rekurzija – što je to?“

U udžbeniku pročitati od 70. do 76. 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):

Rekurzija:

- Pojava u prirodi koja ima svojstvo stalnog ponavljanja (godovi drveta, obrasci kod školjki ili kućica puževa itd.)
- U programiranju je to svaka funkcija koja omogućuje ponavljanje određenih naredbi unutar same funkcije pozivom te funkcije (funkcija poziva samu sebe).

Riješiti vježbe i zadatke i rješenja prepisati u bilježnicu:

Zadatak 11. Nacrtajmo piramidu koja se sastoji od kvadrata tako da se broj kvadrata unutar svakoga sljedećeg reda smanjuje za jedan. Riješimo problem primjenom rekurzivne funkcije.

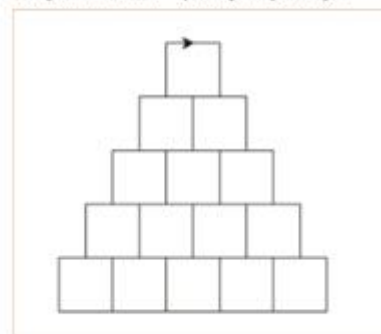
Primjer testiranja

```
>>> piramida(50, 5)
>>>
```

Ovaj smo zadatak rješavali u 7. razredu primjenom petlje **while** pri čemu smo koristili potprograme ili funkcije koje su:

- crtale jedan kvadrat
- crtale niz kvadrata
- postavljale olovku na odgovarajuće mjesto za crtanje sljedećeg niza
- crtale piramidu (glavna funkcija)

Grafički zaslon – primjer rješenja



Dogradimo tu ideju rješenja tako da izmijenimo samo glavnu funkciju koja će sada postati rekurzivna funkcija.

<pre> from turtle import* def kvadrat(a): for i in range(4): fd(a) lt(90) def niz(a,n): for i in range(n): kvadrat(a) fd(a) def postavi(a,n): bk(n*a) lt(90);fd(a);rt(90) fd(a/2) def piramida(a,n): if n<1: return else: niz(a,n) postavi(a,n) piramida(a,n-1) </pre>	funkcija kojom crtamo jedan kvadrat funkcija kojom crtamo niz kvadrata funkcija kojom postavljamo olovku na početak crtanja sljedećeg niza kvadrata glavna (rekurzivna) funkcija uvjet prekida pokretanje funkcije crtanja niza kvadrata pokretanje funkcije pozicioniranja olovke Rekurzivni poziv glavne funkcije uz smanjenje broja kvadrata za 1.
---	---

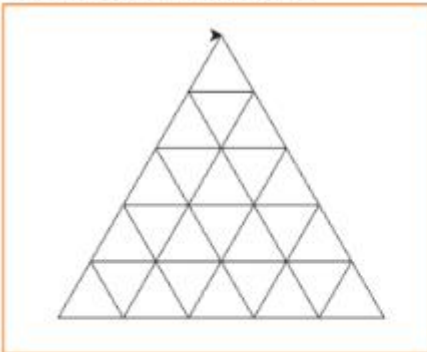
Zadatak 12. Prema prethodnom primjeru nacrtaj piramidu od n jednakostraničnih trokuta duljine stranice a primjenom rekurzivne funkcije.

Primjer testiranja

```
>>> piramida(60,5)
>>>
```



Rezultat u grafičkom zaslonu



Računalni program

```

from turtle import*
def trokut(a):
    for i in range(3):
        fd(a)
        lt(120)
def niz(a,n):
    for i in range(n):
        trokut(a)
        fd(a)
def postavi(a,n):
    bk(n*a)
    lt(60);fd(a);rt(60)
def piramida(a,n):
    if n<1:
        return
    else:
        niz(a,n)
        postavi(a,n)
        piramida(a,n-1)

```

Zadatak 13.

Primijenimo rekurzivnu funkciju koja će izračunati umnožak prvih n prirodnih brojeva.

Ako bismo umnožak prvih n prirodnih brojeva računali bez rekurzivne funkcije, tada bi rješenje uključivalo primjenu petlje **for** kojom se množe svi brojevi do broja n , počevši od broja jedan.

Primjeri testiranja

```
>>> umnozak(4)
24
>>> umnozak(6)
720
>>>
```

Računalni program

```
def umnozak(n):
    u=1
    for i in range(1,n+1):
        u=u*i
    return u
```

Drugi je pristup rješavanju tog problema primjenom rekurzije, odnosno stvaranjem rekurzivne funkcije s definiranim uvjetom zaustavljanja ($n=1$) te načinom daljnjeg pozivanja funkcije ($n*\text{umnozak}(n-1)$). Takvo je računalno rješenje prikazano slikom.

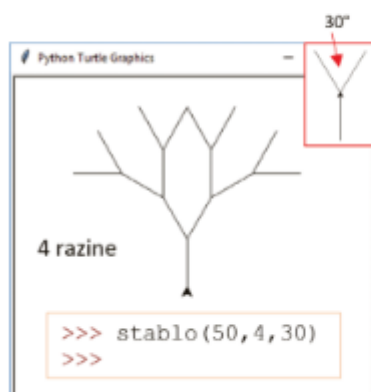
Računalni program

```
def umnozak(n):
    if n==1:
        return 1
    else:
        return n*umnozak(n-1)
```

Uvjet prekida.

Zadatak 14. Nacrtajmo jedno simetrično stablo kojem svaka grana ima točno dvije identične podgrane.

Uočimo na slici osnovni dio stabla koji se ponavlja i nacrtajmo ga u interaktivnom sučelju.



Osnovni dio stabla koji se ponavlja.

Crtanje osnovnog dijela stabla u interaktivnom sučelju

```
>>> from turtle import*
>>> lt(90)
>>> fd(50)
>>> lt(30)
>>> fd(50)
>>> bk(50)
>>> rt(30*2)
>>> fd(50)
>>> bk(50)
>>> lt(30)
>>>
```

Niz programskih naredbi koje smo primijenili za crtanje osnovnog dijela stabla izmijenit ćemo i iskoristiti za crtanje simetričnog stabla. Duljinu jedne grane stabla definirat ćemo varijablom **a**. Simetrično stablo sadržavat će **n** razina, a kut otklona grane definirat ćemo varijablom **kut**. Dakle, umjesto poziva funkcije **stablo (50, 4, 30)** upotrijebit ćemo općeniti poziv funkcije **stablo (a, n, kut)**.

Funkcija **stablo** bit će **rekurzivna**, dakle pozivat će samu sebe. Argument (**n**) (broj razine) će se pri svakom pozivu funkcije umanjiti za jedan (**n-1**). Pozivanje funkcije zaustavit će se u trenutku kada broj grana bude manji od jedan (**n<1**).

Računalni program

```
from turtle import *
lt(90)
def stablo(a,n,kut):
    if n<1:
        return
    else:
        fd(a)
        lt(kut)
        stablo(a,n-1,kut)
        rt(2*kut)
        stablo(a,n-1,kut)
        lt(kut)
        bk(a)
```

Prije početka crtanja stabla olovku treba zaokrenuti tako da gleda prema gore

uvjet zaustavljanja rekurzivnog pozivanja funkcije

rekurzivni poziv crtanja lijeve grane stabla

rekurzivni poziv crtanja desne grane stabla

Primjer testiranja za različite vrijednosti argumenta **n**

<code>stablo(50,1,30)</code>	<code>stablo(50,2,30)</code>	<code>stablo(50,3,30)</code>	<code>stablo(50,4,30)</code>
			



Provjerite svoje znanje

- Možete li navesti neke primjere pojava/stvari iz svakodnevnog života u kojima uočavate rekurzivni postupak?
- Navedite glavna obilježja rekurzivnog postupka.

Riješiti radnu bilježnicu zadatke od 1. do 6. na stranici 24 i 25 (ponekad su to različiti brojevi stranica zbog razlike u izdanju).

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

3.3. Rekurzija – što je to?

1. Sljedeće slike prikazuju neke od rekurzivnih pojava koje možemo susresti u prirodi. Napiši nazive prikazanih rekurzivnih pojava.



2. S pomoću interneta istraži i pronadi još neke rekurzivne pojave te ih na predviđenim crtama navedi i opiši.

3. Svojim riječima opiši što je rekurzija.

4. Sljedeći računalni program prikazuje tipičan primjer primjene rekurzije u programiranju.

```
def umnozak(n):  
    if n==1:  
        return 1  
    else:  
        return n* umnozak(n-1)
```

Taj će računalni program izračunati umnožak prvih n prirodnih brojeva. U programu su istaknuti (uokvireni) njegovi važni dijelovi. Na predviđene crte upiši i objasni njihovo značenje.

5. Sljedeći računalni program prikazuje primjenu rekurzije u programiranju.

Primjer testiranja

```
>>> kvadrat(250,10,45)
```

Računalni program

```
from turtle import *
lt(90)
def kvadrat(a,n,kut):
    if n<1:
        return
    else:
        for i in range(4):
            fd(a)
            lt(90)
            lt(kut)
            kvadrat(a*2/3,n-1,kut)
```

Grafički prikaz računalnog rješenja



Promotri prikazani računalni program. Odgovore na postavljena pitanja napiši na predviđena prazna mjesta.

a) Koliko će se kvadrata nacrtati prikazanim računalnim programom? _____

b) Na koji se način računa duljina svakog sljedećeg (manjeg) kvadrata pri crtanju tog crteža

c) Prije crtanja svakoga sljedećeg kvadrata olovka (kornjača) se okreće za _____ stupnjeva ulijevo.

6. Dopuni računalni program iz prethodnog zadatka tako da se programom umjesto kvadrata crtaju jednakostranični trokuti, pri čemu je duljina svakoga sljedećeg trokuta dvostruko manja.

Primjer testiranja

```
>>> trokut(250,6,45)
```

Računalni program

```
from turtle import *
lt(90)
def trokut(a,n,kut):
    if n<1:
        return
    else:
        for i in range(____):
            fd(a)
            lt(____)
            lt(kut)
            trokut(____,n-1,kut)
```

Grafički prikaz računalnog rješenja

