

GEODETSKA ŠKOLA ZAGREB

Python

Programiranje u Pythonu - bilješke s predavanja

Tihana Polezari

prof. Armando Slaviček



Nastavne cjeline

1. O programiranju

- a. Povijest programiranja
- b. Faze razvoja programa

2. Programski jezici

- a. Podjela programskih jezika
- b. Odabir programskog jezika

3. Algoritmi

- a. Pojam algoritma
- b. Dijagram toka
- c. Simboli dijagrama toka

4. Kodiranje

- a. Pseudokod
- b. Varijable
- c. Operatori
- d. Slijed
- e. Grananje
- f. Petlje
- g. Kompajliranje
- h. Testiranje

5. Izrada dokumentacije

- a. Korisnička dokumentacija
- b. Programerska dokumentacija

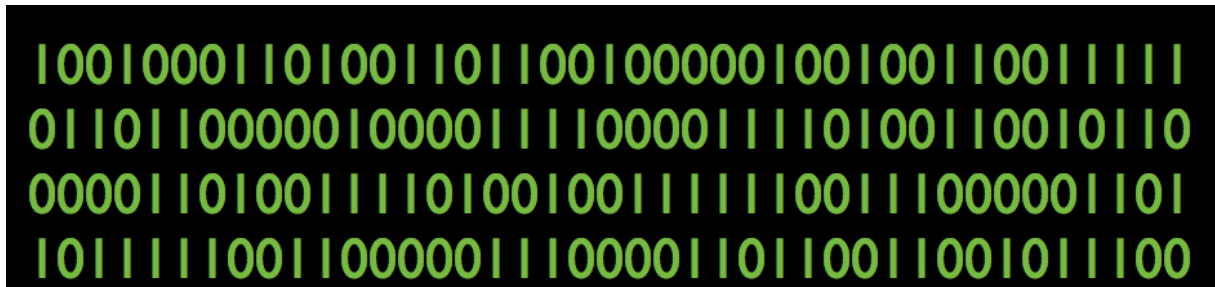
Sadržaj

PROGRAMSKI JEZICI.....	3
Povijest programiranja	3
Faze razvoja programa	7
Odabir programskog jezika.....	7
Što je Python?.....	8
Algoritam	9
OSNOVNI ALGORITAMSKI POSTUPCI	10
Algoritamski zadaci.....	13
Logički ili Booleov tip podataka	15
LOGIČKI OPERATORI	16
PONAVLJANJE:.....	17
PONAVLJANJE	19
FORMATIRANI ISPIS.....	21
Neka osnovna pravila pisanja programa	23
PRAVILA ZA PISANJE IMENA	24
NAREDBE PRIDRUŽIVANJA	25
JEDNOSTAVNI PROGRAMI.....	32
DONOŠENJE ODLUKA I GRANANJA U PROGRAMIMA	33
LOGIČKI OPERATORI I LOGIČKI IZRAZI.....	34
Donošenje odluka u programima.....	40
Moduli	57
Modul random.....	58
Osnove računalne grafike.....	66

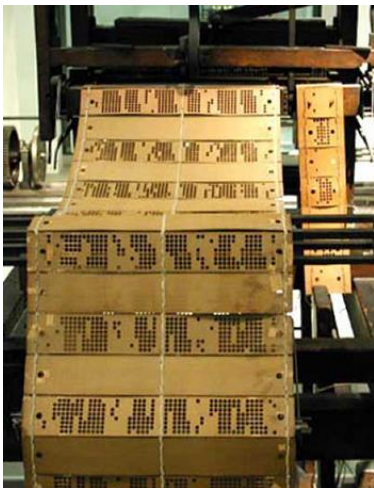
PROGRAMSKI JEZICI

Računala se također sporazumijevaju jezicima. Jezici koje računala razumiju zovu se **programski jezici**.

Možemo reći da je **programski jezik skup** ključnih riječi i pravila za njihovo korištenje koje „razumije“ računalo.



Povijest programiranja



Korijeni u tekstilnoj industriji (1801.)

Francuz Joseph Marie Charles **Jacquard** po zanimanju tkalac izradio je program za tkalački stroj, izrađen na drvenoj bušenoj kartici.

Ideja programiranja

- rašlanjivanje kompleksnih zadataka na niz nedvosmislenih i konačnih koraka koje stroj može izvesti

- stroj na temelju programa može izvršavati ponavljajuće zadatke

Prvi programer/ka:



1842. Ada Lovelace Byron

„Analitički stroj tka algebarske uzorke na isti način kako Jacquard-ov tkalački stroj tka cvjetove i listove. „

Ada je napisala skupove instrukcija koje bi se mogle izvršavati na analitičkom stroju.

Ada – prvi programer za računala. Programski jezik Ada je u njenu čast dobio ime.

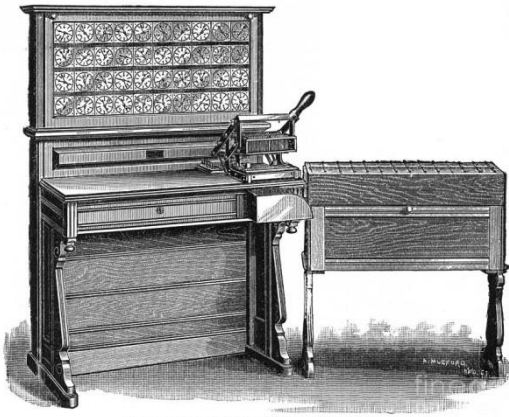
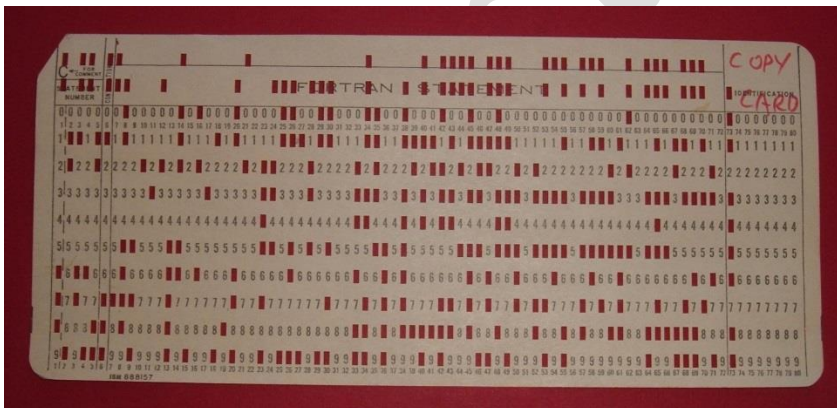
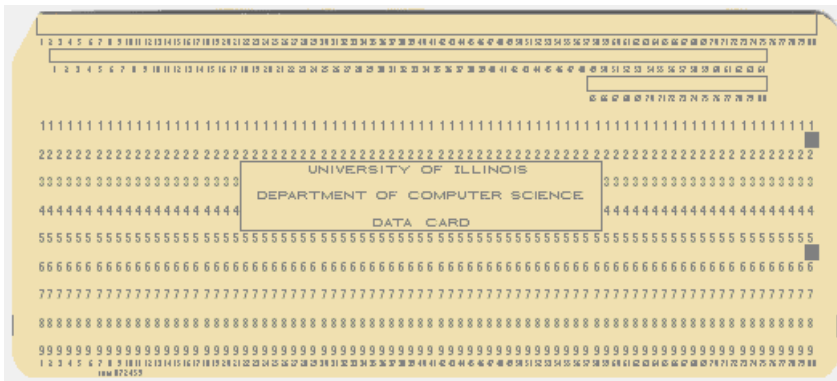


Fig. 5.—The Hollerith electric tabulating machine.

Herman Hollerith (1860 – 1929), njemačko-američki statističar, izumitelj stroja na principu bušenih kartica koji se koristio pri popisu stanovništva Amerike (1890.). Smatra se začetnikom elektromehaničke obrade podataka.

Bušene kartice



Generacije programskih jezika

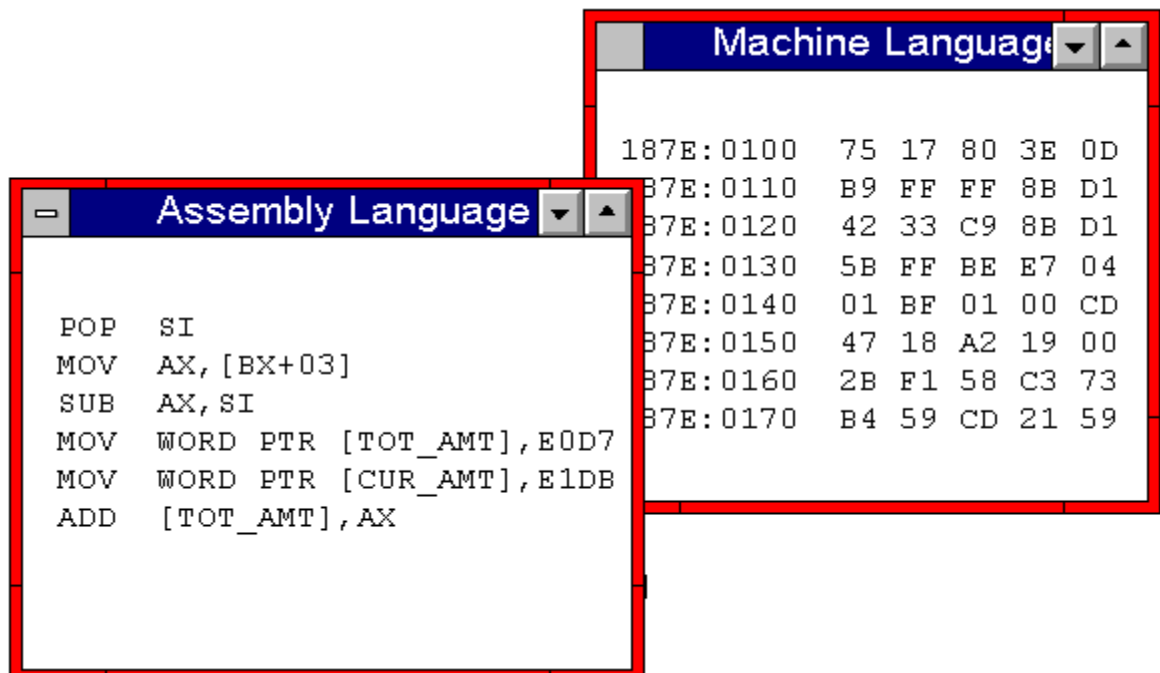
1. Strojni
2. Simbolički - Asembler
3. Proceduralni
4. Problemski orijentirani
5. Prirodni

1. strojni programski jezik

- sastoji se od stringova brojeva (ograničenih na 0 i 1) koji računalu kazuju kako izvršiti elementarne operacije jednu po jednu
- primjer instrukcije:
01110001110011100110011000011

2. simbolički - asemblerski programski jezik

- kratice na engleskom jeziku za prikaz elementarnih operacija (mnemonički kod)
- primjer:
MOVE BX, AX



The image shows two overlapping windows from a software application. The top window is titled "Machine Language" and displays a list of memory addresses and their corresponding hexadecimal values. The bottom window is titled "Assembly Language" and displays the corresponding assembly instructions for each address.

Address	Machine Language	Assembly Language
187E:0100	75 17 80 3E 0D	POP SI
B7E:0110	B9 FF FF 8B D1	MOV AX, [BX+03]
B7E:0120	42 33 C9 8B D1	SUB AX, SI
B7E:0130	5B FF BE E7 04	MOV WORD PTR [TOT_AMT], E0D7
B7E:0140	01 BF 01 00 CD	MOV WORD PTR [CUR_AMT], E1DB
B7E:0150	47 18 A2 19 00	ADD [TOT_AMT], AX
B7E:0160	2B F1 58 C3 73	
B7E:0170	B4 59 CD 21 59	

3. proceduralni programski jezici/viši programski jezici (treća generacija)

- naredba je izraz kojim se izvodi niz operacija na sklopovskoj razini računala
- (LOGO, BASIC, LISP, PROLOG, Ada, Pascal, ...)

<u>PROGRAMSKI JEZIK LOGO</u>	<u>PROGRAMSKI JEZIK QBASIC</u>	<u>PROGRAMSKI JEZIK Pascal</u>
<pre> TO ZBROJI MAKE "A READ MAKE "B READ MAKE "C :A+:B PR :C END </pre>	<pre> INPUT A INPUT B C=A+B PRINT C END </pre>	<pre> Program zboji; var a, b: integer; begin; readln (a); readln (b); c:=a+b; writeln (c); end. </pre>

4. problemski orijentirani programski jezik/ jezici četvrte generacije

- komponente ovih jezika su baze podataka s vlastitim upravljačkim sustavom, rječnik podataka (u rječniku se nalazi detaljan opis svih podataka, izgledi ekrana) te alati koji osiguravaju zaštitu baze podataka
- **Structured Query Language (SQL)** – najpopularniji programski jezik za kreiranje, dohvat, ažuriranje i brisanje podataka iz baze podataka
- **PostScript (PS)** – jezik za opisivanje izgleda stranice i programski jezik koji se prvenstveno koristi u elektroničkom i stolnom izdavaštvu

5. prirodni programski jezik

- C++:C
- JAVA
- VisualBasic
- Generatori
- Razvojna okruženja

Faze razvoja programa

1. Definicija problema
2. Skiciranje rješavanja
3. Sastavljanje algoritma
4. Kodiranje
5. Prevođenje programa
6. Testiranje programa
7. Dokumentiranje
8. Održavanje programa

Odabir programskog jezika

Prostoji više različitih programskih jezika. Oni su izmišljeni da bi čovjeku olakšali komunikaciju s računalom, tj. Programiranje. Budući da bi programiranje bilo preteško kad bismo sami morali prirediti strojni kod, napravljeni su razvojni alati koji služe kao pomoć u pisanju programa. Oni podržavaju razne programerske jezike i pripremaju naše programe za izvođenje na računalu. Neki od najčešće korištenih programa su:

- FORTRAN, COBOL, CLIPPER, LOGO, PASCAL, C, C++, C#, Java, VisualBasic, HTML, CSS, PHP

Što je Python?

- programski jezik visoke razine
- razvoj začet 1991., Guido Van Rossum
- slobodan softver
- interpreter prevodi kod tijekom izvršavanja
- multiplatformalan (izvršava se na windows, linux, macos, ..., platformama)
- jednostavan
- dinamičan
- čista sintaksa – lakše čitanje koda
- velik broj modula i biblioteka
- velike mogućnosti iskorištavanja
- primjena u geoinformatici
- ne donosi neke nove revolucionarne značajke programiranju, već na optimalan način ujedinjuje sve najbolje ideje i načela rada drugih programskih jezika

Bilješke s predavanja

Algoritam

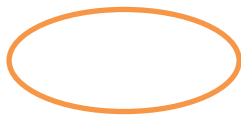
Algoritam se najčešće predstavlja pomoću **dijagrama tijeka** i **pseudojezika**.

Dijagram tijeka je grafički način predstavljanja algoritma skupom grafičkih simbola koji označuju pojedine operacije u algoritmu.

Pseudojezik oponaša sintaksu programskih jezika koristeći izraze, tj. govorni jezik kojim oponašamo naredbe i sintaksu programskih jezika.

Dijagram tijeka

Algoritme za rješavanje problema najčešće pokazujemo grafički pomoću dijagrama tijeka. Dijagram tijeka grafički je prikaz algoritma. Tako prikazan algoritam vrlo je pregledan i potpuno određen. Posebno je pogodan za analize programa, traženje sličnih rješenja ili potrebne izmjene. Pri crtanju dijagrama služimo se posebnim znakovima.



početak



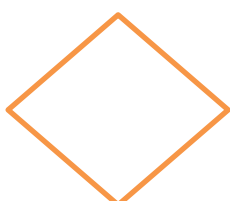
ulaz podataka



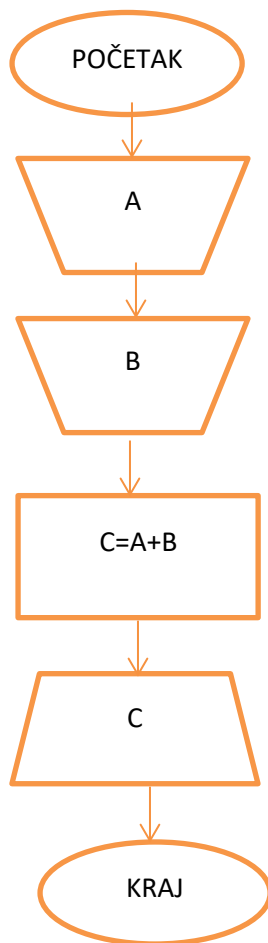
izlaz podataka



naredba



odluka

OSNOVNI ALGORITAMSKI POSTUPCI**SLIJED**

početak
unesi broj A
unesi broj B
 $C = A + B$
Ispiši C
kraj

Bilješke s predavanja

GRANANJE

Početak

Ispis „želite li registrirati program“

Učitaj odgovor

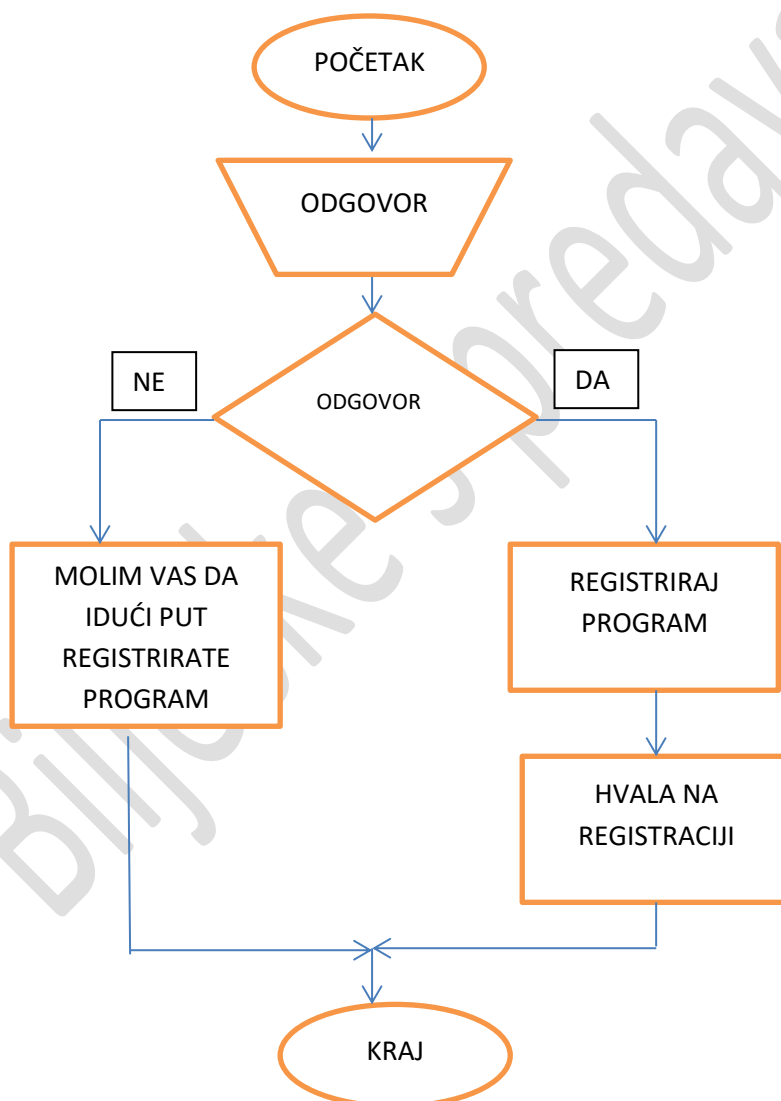
Ako je odgovor „da“ registriraj program

Ispis „hvala na registriranju“

Inače

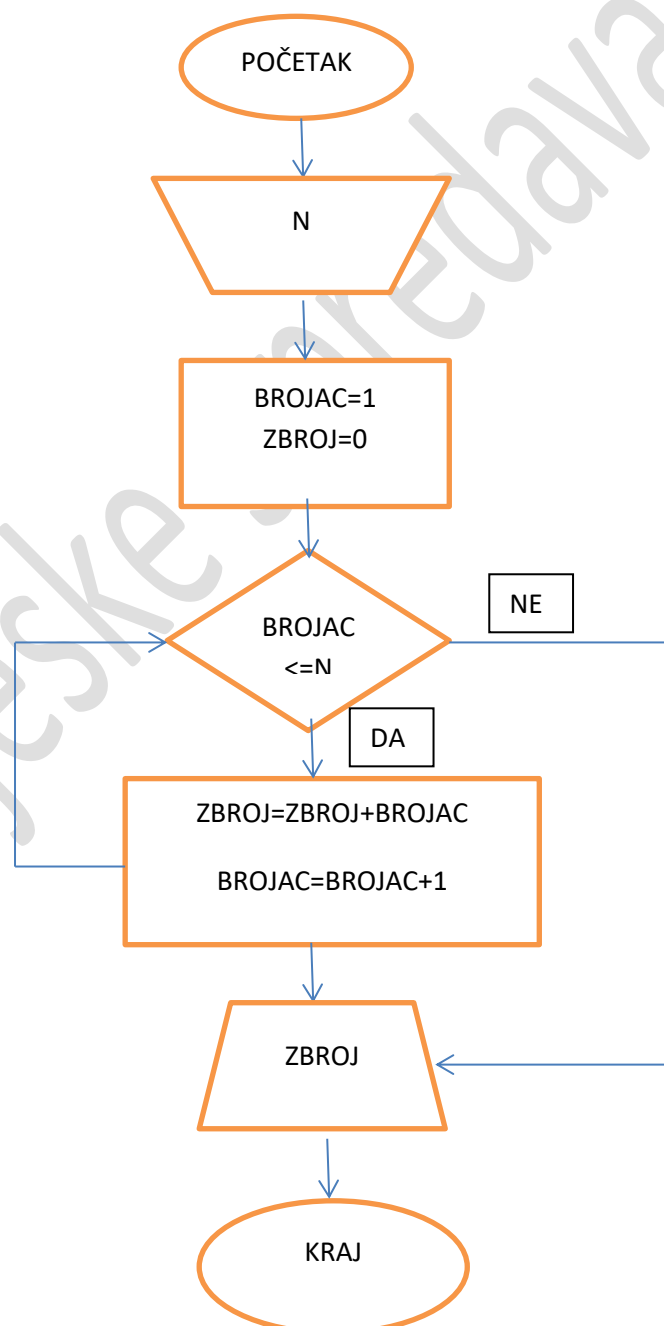
Ispis „molimo da idući put registrirate program“

Kraj



PETLJA

Početak
Učitaj N
Brojac=1
Zbroj=0
Dok je brojac \leq N
Zbroj=zbroj+brojac
Brojac=brojac+1
Ispiši zbroj
Kraj



Algoritamski zadaci

Zadatak 1. Napiši algoritam i dijagram tijeka za izračunavanje zbroja dva broja i ispisivanje dobivenog zbroja.

Pseudokod

Početak

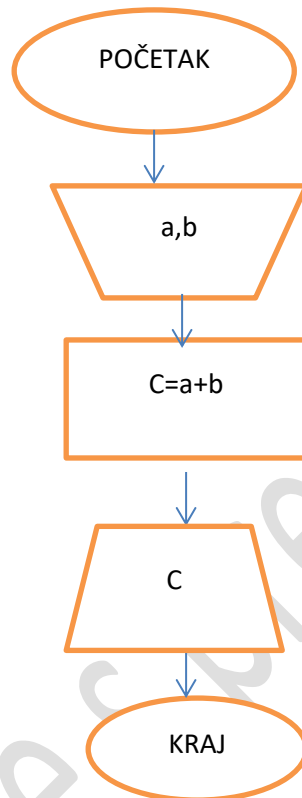
unesi broj a

unesi broj b

$C=a+b$

ispiši C

kraj



```
>>> a=int(input("broj a iznosi="))
broj a iznosi= 5
>>> b=int(input("broj b iznosi="))
broj b iznosi= 6
>>> print("zbroj tih brojeva iznosi=", a+b)
zbroj tih brojeva iznosi= 11
>>>
```

Zadatak 2. Napišimo algoritam i dijagram tijeka za izračunavanje opsega raznostraničnog trokuta, kome su zadane duljine stranica a,b i c.

Pseudokod

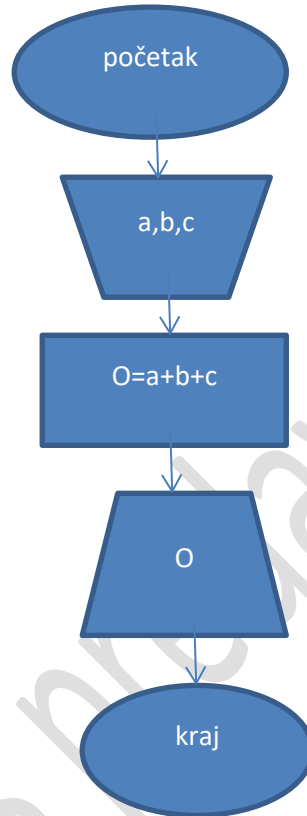
Početak

unesi a,b,c

$O=a+b+c$

ispiši = O

kraj



```
>>> a=int(input("stranica a je velika="))
stranica a je velika= 5
>>> b=int(input("stranica b je velika="))
stranica b je velika=8
>>> c=int(input("stranica c je velika="))
stranica c je velika=9
>>> print("opseg jednakostraničnog trokuta iznosi =", a+b+c)
opseg jednakostraničnog trokuta iznosi = 22
>>>
```

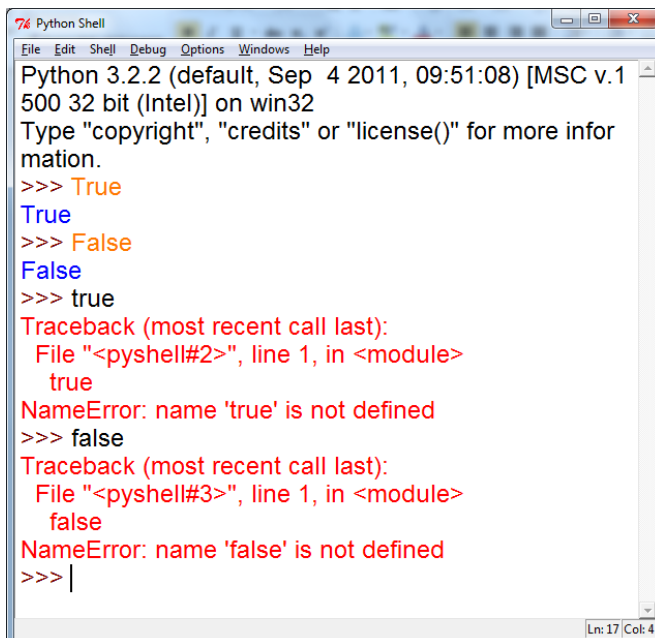
Logički ili Booleov tip podataka

Naziv logički tip potječe iz logike – grane filozofije koja se bavi oblicima ispravne misli i metodama spoznaje. Osnovni pojam u logici je **logički sud**. Pod tim nazivom podrazumijeva se svaka tvrdnja koja se ocjenjuje samo s gledišta **istinitosti** ili **lažnosti**. Neka druga svojstva sudova (primjerice: lijepo, ružno, dobro, loše) u logici se ne razmatraju. logički sudovi moraju biti valjani. Valjani sudovi moraju biti tako oblikovani da mogu biti **istiniti** ili **lažni**.

Za istraživanje sudova i složenih sudova razvijena je posebna grana matematike – matematička logika. Osnovne matematičke logike čini algebra sudova znana po nazivima **logička algebra** ili **Booleova algebra**.

U Pythonu se logički tip naziva **bool**. Taj tip može primiti samo dvije vrijednosti:

True i **False**.



```
Python Shell
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> True
True
>>> False
False
>>> true
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    true
NameError: name 'true' is not defined
>>> false
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    false
NameError: name 'false' is not defined
>>> |
```

Python ima ugrađene funkcije **int()** i **bool()** pomoću kojih možemo ustanoviti cjelobrojnu vrijednost neke logičke vrijednosti i obrnuto, kakva je logička vrijednost neke cjelobrojne vrijednosti:

```
>>> int(True)    Pokazuje se da je cjelobrojni ekvivalent vrijednosti True jednak 1 te
1                vrijednosti False jednak 0.
>>> int(False)
0
>>> bool(1)
True
>>> bool(0)
False
>>>
```


LOGIČKI OPERATORI

Najčešće se koriste tri operatora: **AND**, **NOT** i **OR**.

AND - operator koji je istinit samo ako su uvjeti s obje njegove strane istiniti (1 AND 1).

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

NOT – operator koji pretvara 0 u 1 i obrnuto.

A	NOT A
0	1
1	0

OR – operator koji nije istinit samo ako uvjeti s obje njegove strane nisu istiniti (0 OR 0).

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Vidimo da operator **NOT** uvijek invertira vrijednost (0 postaje 1 i 1 postaje 0), operator **AND** je istinit samo ako su oba parametra istinita, a **OR** je istinit kad god je barem jedan parametar istinit.

PONAVLJANJE:

1. Nabroji generacije programskih jezika
2. Nabroji faze razvoja programa
3. Nabroji neke od najčešće korištenih programa
4. Razvoj Pythona započeo je _____ godine.
5. Algoritam se najčešće predstavlja pomoću _____ i _____
6. Dijagram tijeka je _____ prikaz programa
7. Nacrtaj grafičke simbole i objasni njihovo značenje koji označuju pojedine operacije u algoritmu
8. Osnovni algoritamski postupci su:
9. Napiši pseudokod za unos dva broja i izračunavanje njihove sume i ispis sume.
10. Napiši algoritam i dijagram tijeka za izračunavanje opsega raznostraničnog trokuta, kome su zadane duljine stranica a,b i c.
11. Osnovni tipovi podataka u Pythonu su:
12. Za ispis cijelog broja u binarnom odnosno heksadekadskom obliku postoje funkcije _____ i _____
13. U Pythonu se logički tip naziva _____, koji može poprimiti dvije vrijednosti _____ i _____
14. `int(True) = ?`, `int(False) = ?`
15. Posebni znak za oblikovanje teksta `\t` znači rijelaz u novi redak? Da-ne
16. Naredba `True` i `true` je ista naredba? DA – NE
17. Znakovni nizovi mogu se ograditi ili jednostrukim ili dvostrukim znakovima? DA – NE

Rješavanje problema i stjecanje novih spoznanja računalomKoraci izgradnje programa su:

- analiza zahtjeva,
- specifikacija problema,
- odabir algoritma i zasnivanje programa
- pisanje programa
- provjeravanje i ispitivanje programa
- održavanje programa.

Algoritam - konačan niz koraka koji vodi prema rješenju nekog problema .

Algoritamski proces - obavljanje programa

Algoritam može izvoditi čovjek, životinja ili neki uređaj.

Izvori algoritma su:

- Praktično iskustvo-imitacijski algoritmi,
- Znanstvena teorija- teorijski algoritmi,
- Skup postojećih algoritma- konstrukcijski algoritmi,
- Domišljatost stvaratelja.

Svojstva algoritma su:

- Ispravnost (korektnost)- radi ispravno za sve dopuštene ulaze,
- Trajanje-broj osnovnih operacija od kojih se algoritam sastoji.

Dva su algoritma ekvivalentna ako su:

- Dopuštanje klase objekata i za jedan i za drugi algoritam jednake,
- Završna stanja jednog i drugog algoritma jednake za jednaka početna stanja.

PONAVLJANJE

Zadatak 1.

Unesi stranicu a i b pravokutnika i izračunaj površinu i opseg pravokutnika.

```
>>> a=int(input("stranica a iznosi="))
stranica a iznosi=45
>>> b=int(input("stranica b iznosi="))
stranica b iznosi=12
>>> P=a*b
>>> P
540
>>> O=2*a+2*b
>>> O
114
>>>
```

Zadatak 2.

Unesi radius kruga i izračunaj opseg i površinu kruga.

```
>>> r=float(input("unesi radius r="))
unesi radius r=12.5
>>> pi=float(input("unesi pi="))
unesi pi=3.14
>>> P=r*r*pi
>>> P
490.625
>>> O=2*r*pi
>>> O
78.5
>>>
```

Zadatak 3.

Unesi četiri broja i ispiši aritmetičku sredinu.

```
>>> b1=int(input("stranica a iznosi="))
stranica a iznosi=12
>>> b=24
>>> c=45
>>> d=5
>>> aritsred=(a+b+c+d)/4
>>> print("aritmetička sredina četiri broja=", aritsred)
aritmetička sredina četiri broja= 29.75
>>>
```

Zadatak 4.

Unesi jedan broj od 20. Provjeri dali je veći od 10 ili manji ispiši komentar

(„broj je veći ili broj je manji“).

```
>>> broj=int(input("unesi jedan broj do broja 20="))
unesi jedan broj do broja 20=9
>>> if broj>10:
    print("broj je veci od 10")
else:
    print("broj je manji od 10")

broj je manji od 10
>>>
```

Bilješke s predavanja

FORMATIRANI ISPIS

Formatirani ispis izuzetno je pogodan kada jednim ispisom želimo ispisati više varijabli popraćene odgovarajućim tekstom.

Funkcija **print()** omogućuje precizniji ispis, a sastoji se od varijabli i stringova.

Oblik pisanja:

```
Print('st_1{br_1},st_2{br_2},...,st_n{br_n}',format(v_1,v_2,...,v_n))
```

```
Print('st1{br1},st2{br2},...,stn{brn}',format(v1,v2,...vn))
```

St_1,st_2,...,st_n – proizvoljan tekst

V_1, v_2,..., v_n – popis varijabli koje želimo ispisati

Br_1, br_2,..., br_n – predstavljaju redne brojeve varijabli unutar popisa

```
>>> print('zbroj brojeva {0} i {1} je: {2}'.format(4,5,4+5))
zbroj brojeva 4 i 5 je: 9
>>> print('brojevi zapisani redoslijedom su {3}, {2}, {1}, {0}'.format(5,4,3,2))
brojevi zapisani redoslijedom su 2, 3, 4, 5
>>> print('brojevi zapisani redoslijedom su {0}, {1}, {2}, {3}'.format(5,4,3,2))
brojevi zapisani redoslijedom su 5, 4, 3, 2
```

```
>>> print('zbroj brojeva {1} i {2} je {3}, a {2} i {1} je {3}' .format(25, 35,
25+35, 35+25))
zbroj brojeva 35 i 60 je 60, a 60 i 35 je 60
>>> print('zbroj brojeba {0} i {1} je isto što {1} + {0} = {2}'.format(25,35, 25
+35))
zbroj brojeba 25 i 35 je isto što 35 + 25 = 60
>>> |
```

```
>>> print('zbroj brojeva {1} i {2} je {3}, a {2} i {1} je {3}' .format(25, 35,
25+35, 35+25))
zbroj brojeva 35 i 60 je 60, a 60 i 35 je 60
>>> print('zbroj brojeba {0} i {1} je isto što {1} + {0} = {2}'.format(25,35, 25
+35))
zbroj brojeba 25 i 35 je isto što 35 + 25 = 60
>>>
>>> print('zbroj brojeva {} i {} je: {}'.format(127,423,127+423))
zbroj brojeva 127 i 423 je: 550
>>> print('zbroji {0} i {1} je: {4}, a {2} i {3} je: {5}'.format(25,47,52,117,
25+47, 52+117))
zbroji 25 i 47 je: 72, a 52 i 117 je: 169
>>>
```

Unutar vitičastih zagrada brojeve možemo izostaviti ako se podrazumijeva da podaci idu uzlaznim redom.

Ukoliko želimo pobliže označiti koji tip podataka ispisujemo i u tom ćemo slučaju unutar vitičastih zagrada iza rednog broja varijable napisati dvotočku i nakon toga oznaku tipa:

- d – ako ispisujemo cijeli broj
- f – ako ispisujemo realni broj
- s – ako ispisujemo string

```
>>> print('{2s} brojeva {0:d} i {1:d} je {3:f}'.format(3,4, 'Količnik', 3/4))  
Količnik brojeva 3 i 4 je 0.75000
```

Bilješke s predavanja

Neka osnovna pravila pisanja programa

Tekst u editonu (IDLE sučelja) obojen je različitim bojama tako da se lakše prepozna karakteristični, jezični elementi i održavala pravila programa

Zadatak 1. Napiši program koji će učitati ime korisnika i broj njegovih godina. Ispiši ime, broj godina i broj dana života korisnika.

```
>>> ime=input("kako se zoveš?")
kako se zoveš? Tihana
>>> godina=int(input("koliko imas godina?"))
koliko imas godina? 16
>>> print(ime, "imaš", godina, "a to iznosi", 365*16, "dana")
Tihana imaš 16 a to iznosi 5840 dana
>>> print(ime, "imaš", godina, "a to iznosi", 365 * godina, "dana")
Tihana imaš 16 a to iznosi 5840 dana
>>> print(ime, "imaš", godina, "godina, a to iznosi", 365 * godina, "dana")
Tihana imaš 16 godina, a to iznosi 5840 dana
```


Zadatak 2. Napiši program koji će učitati veličinu kuta u stupnjevima, a zatim stupnjeve pretvoriti u radijane.

```
>>> stupnjevi=int(input("unesi stupnjeve"))
unesi stupnjeve 35
>>> minute=int(input("unesi minute"))
unesi minute45
>>> sekunde=int(input("unesi sekunde"))
unesi sekunde37
>>> dstupnjeva=stupnjevi + minute/60 + sekunde/3600
>>> radijan=dstupnjeva*3.14159/180
>>> print("kut ima", dstupnjeva, "stupnjeva")
kut ima 35.76027777777778 stupnjeva
>>> print("kut ima", radijan, "radijana")
kut ima 0.6241340614660494 radijana
>>>
```

PRAVILA ZA PISANJE IMENA

U Pythonu se imena sastoje od proizvoljnog broja slova., znamenki ili podvlaka, s tim da ne smiju početi znamenkom. Python ima posebna imena koja počinju s dvije podvlake i to treba izbjegavati. Najbolje je da odabrana imena počinju slovom.

Python razlikuje velika i mala slova tako da u a i A dva različita imena.

Python upotrebljava neke riječi zovemo ih **ključnim riječima** za svoje svrhe i korisnik ih ne smije upotrebljavati kao imena. Neka od tih imena su: **int, float, bool, str, True, False.**

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	it	or	yield
assert	else	import	pass	
break	except	in	raise	

Navodimo nekoliko valjanih imena:

x, X, x_1, X_1, x_2, X_2, temperatura, T, broj_cipela, Razred

pri čemu x i X, x_1 i X_1 te x_2 i X_2 nisu parovi istih imena/riječ od šest međusobno različitih imena).

Sljedeća imena nisu valjana:

x.1, Y(2), kamata_u_Š jer se u imenu upotrebljavaju ostali znakovi

1_x, 3_kat, 0_vrijednost jer je prvi znak u imenu znamenka

del, or, is jer su to ključne riječi.

Iako se sastoji od niza znakova, ime nije znakovni niz, odnosno string. Ustanovili smo da je string tip podataka. Poslije ćemo naučiti da imena imaju i druge važne funkcije u Pythonu, no jedna od najvažnijih uloga je pohranjivanje vrijednosti kako bi se vrijednost kasnije mogla upotrijebiti.

Posebno treba istaknuti da Python upotrebljava za kodiranje znakova Unicode i da su naši dijakritički znakovi dopušteni. Prema tome, u imenima možemo upotrebljavati sve naše znakove, što kod drugih jezika nije moguće. Iako se možemo koristiti i našim znakovima, preporučeno je, posebice za imena klasa, funkcija, metoda itd., koristiti se područjem koji je definirano ASCII kodom.

Postavlja se pitanje kako odabrati imena. Kraća imena su jednostavnija i brže ih zapisujemo, ali nakon nekog vremena teško ćemo se sjetiti što nam ona predstavljaju.

Dulja imena zahtijevaju više tipkanja, ali će nas i nakon duljeg vremena podsjećati što nam ona predstavljaju. Preporučljivo je izabrati imena koja će nam olakšati uporabu programa kroz neko dulje vrijeme.

NAREDBE PRIDRUŽIVANJA

Opći oblik naredbe produživanja je:

```
varijabla = izraz
```

pri čemu varijabla mora biti ime odabrano u skladu s pravilima oblikovanja imena u Pythonu.

Znak = ovdje ima značenje pridruživanja, pa ga zovemo **znakom pridruživanja**.

Vrijednost koja se piše desno od znaka = bit će pridružena varijabli čije smo ime naveli s lijeve strane znaka pridruživanja

Primjeri:

```
>>> x=100
>>> x
100
>>> X
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    X
NameError: name 'X' is not defined
>>> x
100
>>> A=1000
>>> A
1000
>>> B=10
>>> C=A+B
>>> C
1010
>>> C=A/B
>>> C
100.0
>>> C=A//B
>>> C
100
>>> X=10
>>> Y=100
>>> Z=1000
>>> X=X+1
>>> X
11
>>> Y=Y+X
>>> Y
111
>>> Z=Z+Y
>>> Z
1111
>>> baza=10
>>> dvostruko=2*baza
>>> dvostruko
20
>>> baza=100
>>> dvostruko
20
>>> dvostruko=2*baza
>>> dvostruko
200
>>>
```

NAREDBE VIŠESTRUKOG PRIDRUŽIVANJA

```
>>> a=100
>>> b=15
>>> zbroj, razlika = a+b, a-b
>>> zbroj
115
>>> razlika
85
>>>
```

```
>>> a=100

>>> b=15

>>> zbroj, razlika = a+b, a-b

>>> zbroj

115

>>> razlika

85

>>>

>>> količnik, ostatak = a//b, a%b

>>> količnik

6

>>> ostatak

10

>>>
```

```
>>>zbroj=15

>>> razlika=17

>>> print('zbroj={} razlika={}'.format(zbroj, razlika))

zbroj=15 razlika=17

>>> količnik=15

>>> ostatak=10

>>> print('količnik={} ostatak = {}'.format(količnik,razlika))

količnik=15 ostatak = 17

>>> print('količnik={}\nostatak={}'.format(količnik,ostatak))

količnik=15

ostatak=10

>>>
```

ZAMJENA VRIJEDNOSTI VARIJABLI

```
>>> a=100

>>> b=10

>>> c=a

>>> a=b

>>> b=c

>>> print(a,b)

10 100

>>>
```

NAREDBA VIŠESTRUKOG PRIDJELJIVANJA

```
>>> x=100
>>> y=10
>>> x,y=y,x
>>> print(x,y)
10 100
>>> x=1
>>> y=2
>>> z=3
>>> x,y,z=z,x,y
>>> print(x,y,z)
3 1 2
>>>
```

PONAVLJANJE

1. Napiši program koji će unositi stranicu a i b pravokutnika i ispisivati površinu i opseg pravokutnika.
2. Napiši program koji će unositi prirodni broj n i ispisivati površinu kvadrata čija stranica ima duljinu n.
3. Cijena računala iznosi n kuna. Ti imaš m kuna ($m < n$). Napiši program koji će unositi cijenu računala n i iznos m koji ti imaš a ispisivati iznos koji je još potreban za kupnju računala.

4. Napiši program koji će unositi dva prirodna broja a i b te ispisivati njihov zbroj, razliku, umnožak i količnik (drugi je broj uvijek različit od nule). Ispis programa treba biti „punog“ oblika, primjerice, za unos brojeva 5 i 7 te operaciju + ispis treba biti 5+7=12.

```
>>> a=int(input("broj a iznosi="))
broj a iznosi= 5
>>> b=int(input("broj b iznosi="))
broj b iznosi= 7
>>> print(a,"+", b, "=", a+b)
5 + 7 = 12
>>> print(a,"-", b, "=", a-b)
5 - 7 = -2
>>> print(a,"/", b, "=", a/b)
5 / 7 = 0.7142857142857143
>>> print(a,"*", b, "=", a*b)
5 * 7 = 35
>>>
```

5. Napiši program koji će unositi iznos odobreno g potrošačkog kredita c, godišnju kamatnu stopu p, broj mjeseci m, a ispisivati kamate prema formuli: $k=cp(m+a)/2400$
primjer:
unos:1000, 8, 12
ispis: 43.33

```
>>> c=int(input("iznos odobrenog potrošačkog kredita iznosi="))
iznos odobrenog potrošačkog kredita iznosi= 1000
>>> p=int(input("godišnja kamatna stopa iznosi="))
godišnja kamatna stopa iznosi= 8
>>> m=int(input("broj mjeseci iznosi="))
broj mjeseci iznosi=12
>>> print("kamate iznose=", c*p*(m+1)/2400)
kamate iznose= 43.333333333333336
>>>
```

6. Ivica ima n kuna za koje želi kupiti prijateljima čokolade. Jedna čokolada stoji m kuna. Ivicu zanima koliko će najviše čokolada moći kupiti te koliko će mu novca nakon toga preostati. Pomogni Ivici i napiši program koji će unositi iznos novca kojim Ivica raspolaže te cijenu jedne čokolade, a ispisivati koliko maksimalno čokolada Ivica može kupiti te koliko će mu novca nakon toga ostati.

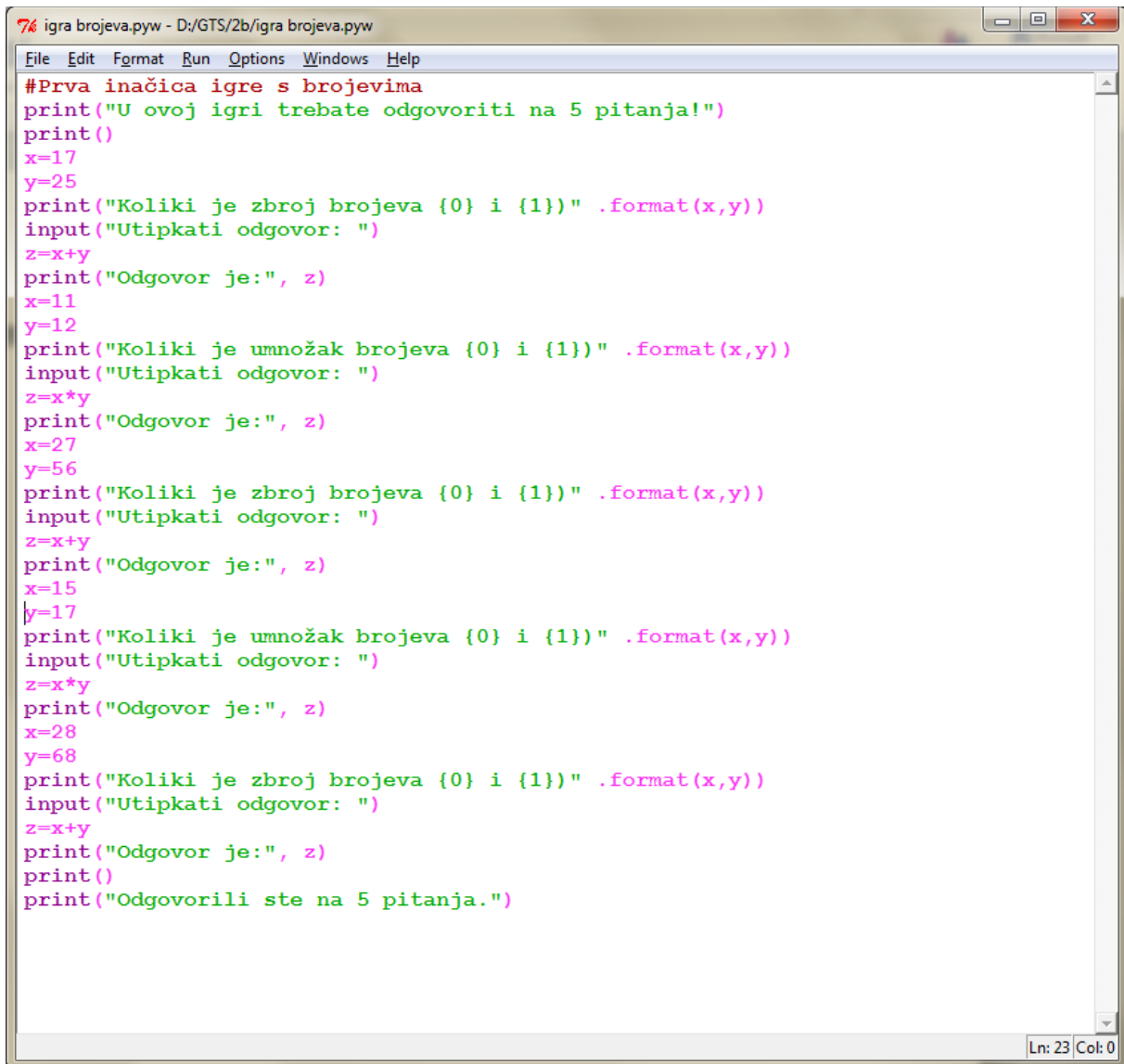
```
>>> n=int(input("ivica ima="))
ivica ima=12
>>> m=int(input("jedna čokolada košta="))
jedna čokolada košta=8
>>> cokolada=n//m
>>> print("ivica može kupiti",cokolada,
"cokolada")
ivica može kupiti 1 cokolada
>>> ostatak=n%m
>>> print("ivici ce ostati", ostatak, "kuna")
ivici ce ostati 4 kuna
>>>
```

7. Napiši program koji će unositi dva prirodna broja n i m . ispisivati koji je veći, a koji manji.

```
>>> n=int(input("broj n je="))
broj n je= 10
>>> m=int(input("broj m je="))
broj m je= 12
>>> if n>m:
    print(n, "je veći od", m)
else:
    print(n, "je manji od", m)
10 je manji od 12
>>>
```


JEDNOSTAVNI PROGRAMI

Zamišljena igra brojevima (1):



```
74 igra brojeva.pyw - D:/GTS/2b/igra brojeva.pyw
File Edit Format Run Options Windows Help
#Prva inačica igre s brojevima
print("U ovoj igri trebate odgovoriti na 5 pitanja!")
print()
x=17
y=25
print("Koliki je zbroj brojeva {0} i {1}" .format(x,y))
input("Utipkati odgovor: ")
z=x+y
print("Odgovor je:", z)
x=11
y=12
print("Koliki je umnožak brojeva {0} i {1}" .format(x,y))
input("Utipkati odgovor: ")
z=x*y
print("Odgovor je:", z)
x=27
y=56
print("Koliki je zbroj brojeva {0} i {1}" .format(x,y))
input("Utipkati odgovor: ")
z=x+y
print("Odgovor je:", z)
x=15
y=17
print("Koliki je umnožak brojeva {0} i {1}" .format(x,y))
input("Utipkati odgovor: ")
z=x*y
print("Odgovor je:", z)
x=28
y=68
print("Koliki je zbroj brojeva {0} i {1}" .format(x,y))
input("Utipkati odgovor: ")
z=x+y
print("Odgovor je:", z)
print()
print("Odgovorili ste na 5 pitanja.")
Ln: 23 Col: 0
```

Ovakav program ima niz nedostataka:

- Nemamo mogućnost uspoređivanja točnog rezultata
- Iste naredbe moramo prepisivati
- Nemamo mogućnost upita igrača želi li igrati dalje
- Igru možemo pokrenuti ponovno pokretanjem programa, ali će se ponoviti zadaci

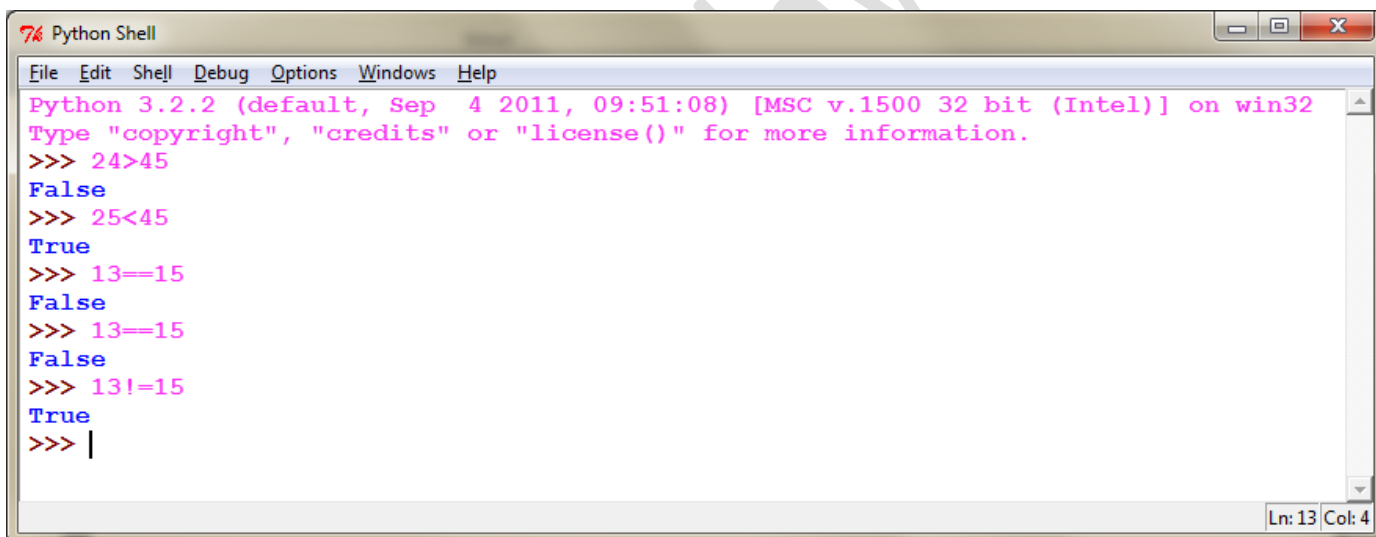
DONOŠENJE ODLUKA I GRANANJA U PROGRAMIMA

Relacijski operatori (operatori uspoređivanja)

Relacijski operatori uspoređuju dva operanda. Rezultat usporedbe ima vrijednosti True ili False.

Prema tome, izraz usporedbe koji se sastoji od dva operanda i relacijskog operatora je logički sud koji može biti istinit ili lažan.

Operator	Značenje simbola
>	Veće od
<	Manje od
>=	Veće od ili jednako
<=	Manje od ili jednako
==	Jednako
!=	Nije jednako



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 24>45
False
>>> 25<45
True
>>> 13==15
False
>>> 13==15
False
>>> 13!=15
True
>>> |
```

#1. Želimo li znati je li neki broj x veći od ili jednak nekoj donjoj granici x_d i istovremeno manji ili jednak nekoj gornjoj granici x_g , primjerice za zadane vrijednosti:

$x_d=0$, $x_g=1000$, $x=700$ ili za $x=1100$.

```
>>> x_d=0
>>> x_g=1000
>>> x=700
>>> x>=x_d
True
>>> x<=x_g
True
>>> x=1100
>>> x>=x_d
True
>>> x<=x_g
False
```

LOGIČKI OPERATORI I LOGIČKI IZRAZI

Za stvaranje bilo kojeg složenog suda dovoljna su nam 3 operatora.

Operator	Naziv operacije	Algebarski simboli
and	I operacija, konjugacija	\wedge
or	II operacija, disjunkcija	\vee
not	NE operacija, komplementiranje	\neg

```
>>> False and False
```

```
False
```

```
>>> False and True
```

```
False
```

```
>>> True and False
```

```
False
```

```
>>> True and True
```

```
True
```

Prema tome, I operacija daje rezultat True samo onda kada oba operatora imaju vrijednost True.

Umjesto vrijednosti False i True mogu se upotrebljavati vrijednosti 0 i 1.

Bilješke s predavanja

```

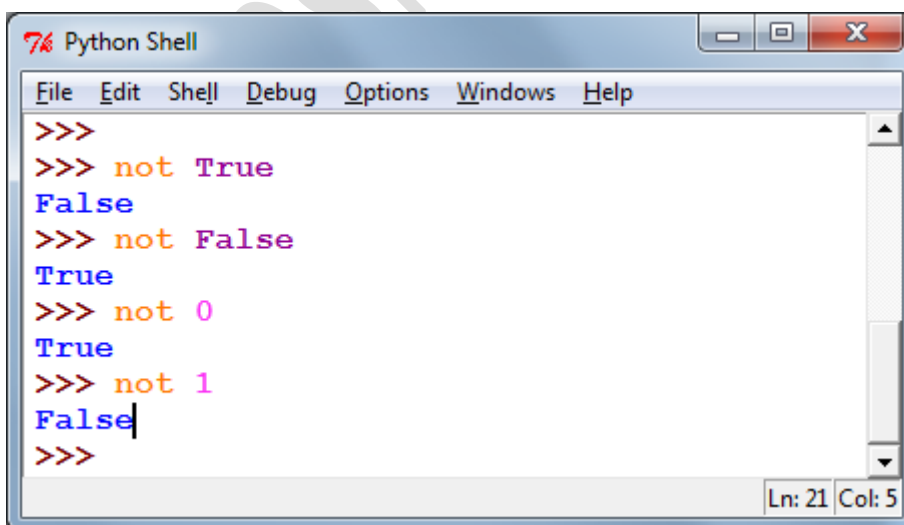
>>> 0 and 0
0
>>> 0 and 1
0
>>> 1 and 0
0
>>> 1 and 1
1
>>>

```

DJELOVANJE OPERATORA OR:

<pre> >>> False or False False >>> False or True True >>> True or False True >>> True or True True </pre>	<pre> >>> 0 or 0 0 >>> 0 or 1 1 >>> 1 or 0 1 >>> 1 or 1 1 >>> </pre>
---	---

Operator negacije NOT djeluje na jedan operand i invertira njegovu vrijednost.



```

Python Shell
File Edit Shell Debug Options Windows Help
>>>
>>> not True
False
>>> not False
True
>>> not 0
True
>>> not 1
False
>>>
Ln: 21 Col: 5

```

Nedosljednost u jeziku Python.

```
>>> not 0
True
>>> not 1
False
>>>
```

Prioritet izvođenja logičkih operacija

Redosljed	Operacija
1.	not
2.	and
3.	or

```
>>> True or False and True
True
>>> False or True and False
False
>>> False or not False
True
>>>
```

Redosljed	Operacija
1.	aritmetički
2.	relacijski
3.	logički

```
>>> 2+3<4
False
>>> 150+150<290
False
>>> 2+3-1>3*3-5
False
>>> 2*3-1>3*3-5
True
>>> 3+4>5 and 3+5>4 and 4+5>3
True
>>> 3+4>5 and 3+5>4 and 4+5>13
False
>>>
```

Ispitivanje je li broj x veći od ili jednak donjoj granici x_d i istovremeno manji od ili jednak gornjoj granici x_g može se provesti s ovim složenim uvjetom:

$(x_d \leq x) \text{ and } (x \leq x_g)$

Uvjet će biti istinit ako se vrijednost varijable x nalazi u zadanim granicama.

$x_d \leq x \text{ and } x \leq x_g$

isti se uvjet može napisati na sljedeći način:

$x_d \leq x \leq x_g$

```
>>> xd=0
>>> xg=1000
>>> x=750
>>> (x>=xd) and (x<=xg)
True
>>> x>=xd and x<=xg
True
>>> x=1500
>>> x>=xd and x<=xg
False
>>> x=-5
>>> x>=xd and x<=xg
False
>>>
```

Bilješke s predavanja

Donošenje odluka u programima

Pseudojezik (nije riječ o stvarnom programskom jeziku)

...

Ako je uvjet onda

```
{  
    Naredba 1_1;  
    ...  
    Naredba 1_n;  
}
```

inače

```
{  
    Naredba 2_1  
    ...  
    Naredba 2_m  
}
```

...

U **Pythonu** se odabir alternativnih blokova naredbi obavlja naredbama u kojima se rabe sljedeće ključne riječi: `if`, `else` i `elif` iza kojih se stavlja dvotočka.

Blokovi naredbi ne ograđuju se posebnim simbolima.

Iza zadnje naredbe bloka mora pisati naredba koja nije uvučena. Blok je na taj način jasno grafički određen u tekstu programa.

...

```
if uvjet:
    naredba 1_1
    ...
    naredba 1_n
else:
    naredba 2_1
    ...
    naredba 2_m
```

Niz naredbi koje smo označili sa: naredba 1_1, naredba 1_2,..., naredba 1_n zvat ćemo blok naredbi i on će se izvesti ako je uvjet istinit. Primijetimo da je isto tako niz naredbi naredba 2_1, naredba 2_2,..., naredba 2_m jedan blok naredbi koji se izvodi ako je uvjet lažan.

Na osnovi navedenog slijedi opći oblik naredbe `if` koji zapisujemo na slijedeći način:

```
...
if uvjet:
    blok_naredbi_1
else:
    blok_naredbi_2
...
```

Zadatak: Napišimo program koji će zahtijevati utipkavanje jednog prirodnog broja i zatim ispisivati je li taj broj djeljiv sa sedam.

```

7% *broj djeljiv sa 7.pyw - D:/GTS/2b/programiranje/broj djeljiv sa 7.pyw*
File Edit Format Run Options Windows Help
#Napišimo program koji će zahtijevati utipkavanje jednog prirodnog
#broja i zatim ispisivati je li taj broj djeljiv sa 7.

print("Program će ispisati je li neki utipkani djeljiv sa 7")
broj=int(input("utipkaj prirodan broj:"))
if broj % 7 ==0:
    print("broj {} je djeljiv sa 7.".format(broj))
else:
    print("Broj {} nije djeljiv sa 7.".format(broj))
|
Ln: 11 Col: 0

```

```

7% Python Shell
File Edit Shell Debug Options Windows Help
False
>>> ----- RESTART -----
>>>
Program će ispisati je li neki utipkani djeljiv sa 7
utipkaj prirodan broj:49
broj 49 je djeljiv sa 7.
>>> ----- RESTART -----
>>>
Program će ispisati je li neki utipkani djeljiv sa 7
utipkaj prirodan broj: 50
Broj 50 nije djeljiv sa 7.
>>> broj
50
>>> 50%7
1
>>> |
Ln: 70 Col: 4

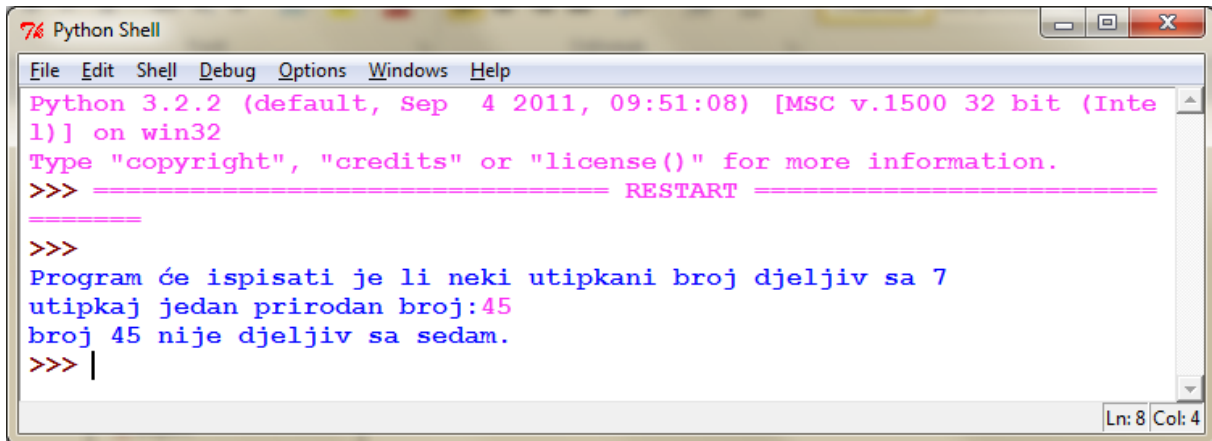
```

Zadatak:

```

7% broj djeljiv sa 7 (2).pyw - D:/GTS/2b/programiranje/broj djeljiv sa 7 (2).pyw
File Edit Format Run Options Windows Help
#Broj je djeljiv sa 7
print("Program će ispisati je li neki utipkani broj djeljiv sa 7")
broj=int(input("utipkaj jedan prirodan broj:"))
if broj %7==0:
    glagol= "je"
else:
    glagol= "nije"
print("broj {} {} djeljiv sa sedam.".format(broj, glagol))
|
Ln: 9 Col: 0

```



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep  4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Program će ispisati je li neki utipkani broj djeljiv sa 7
utipkaj jedan prirodan broj:45
broj 45 nije djeljiv sa sedam.
>>> |
```

Uporabe ključne riječi elif:

-elif za izgradnju programskih struktura višestrukih izbora

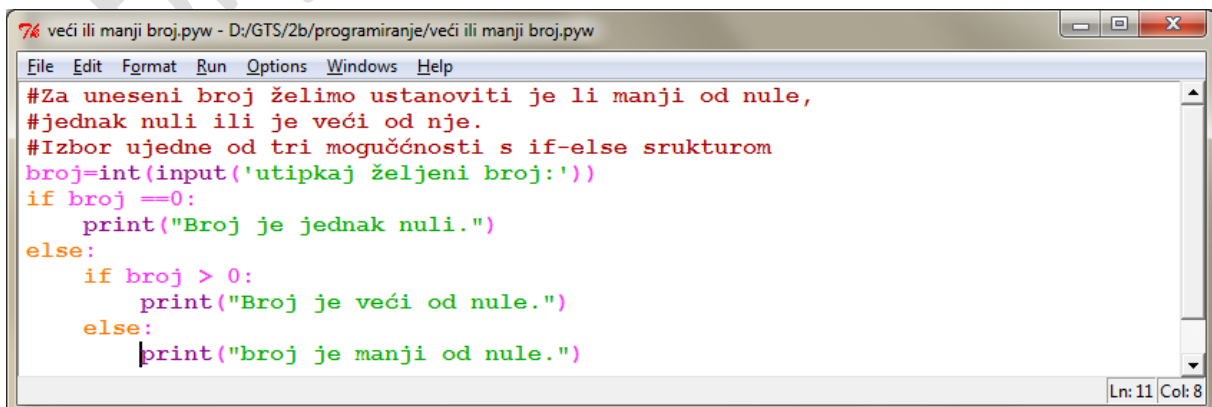
Način izgradnje takve strukture može se opisati na sljedeći način:

...

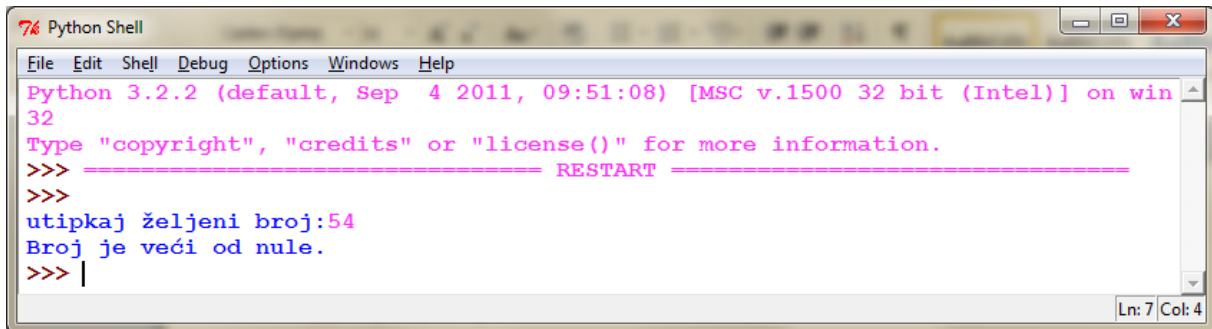
<pre>If logički izraz 0: blok naredbi 0 elif logički izraz 1 blok naredbi 1 elif logički izraz 2 blok naredbi 2 ... elif logički izraz n: blok naredbi n else: blok naredbi</pre>

...

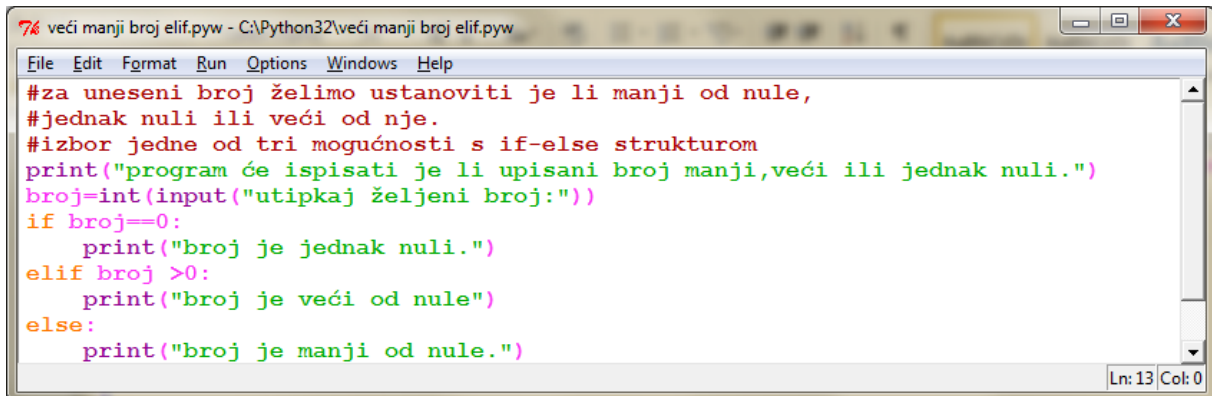
Zadatak:



```
veći ili manji broj.pyw - D:/GTS/2b/programiranje/veći ili manji broj.pyw
File Edit Format Run Options Windows Help
#Za uneseni broj želimo ustanoviti je li manji od nule,
#jednak nuli ili je veći od nje.
#Izbor ujedne od tri mogućnosti s if-else strukturom
broj=int(input('utipkaj željeni broj:'))
if broj ==0:
    print("Broj je jednak nuli.")
else:
    if broj > 0:
        print("Broj je veći od nule.")
    else:
        print("broj je manji od nule.")
```

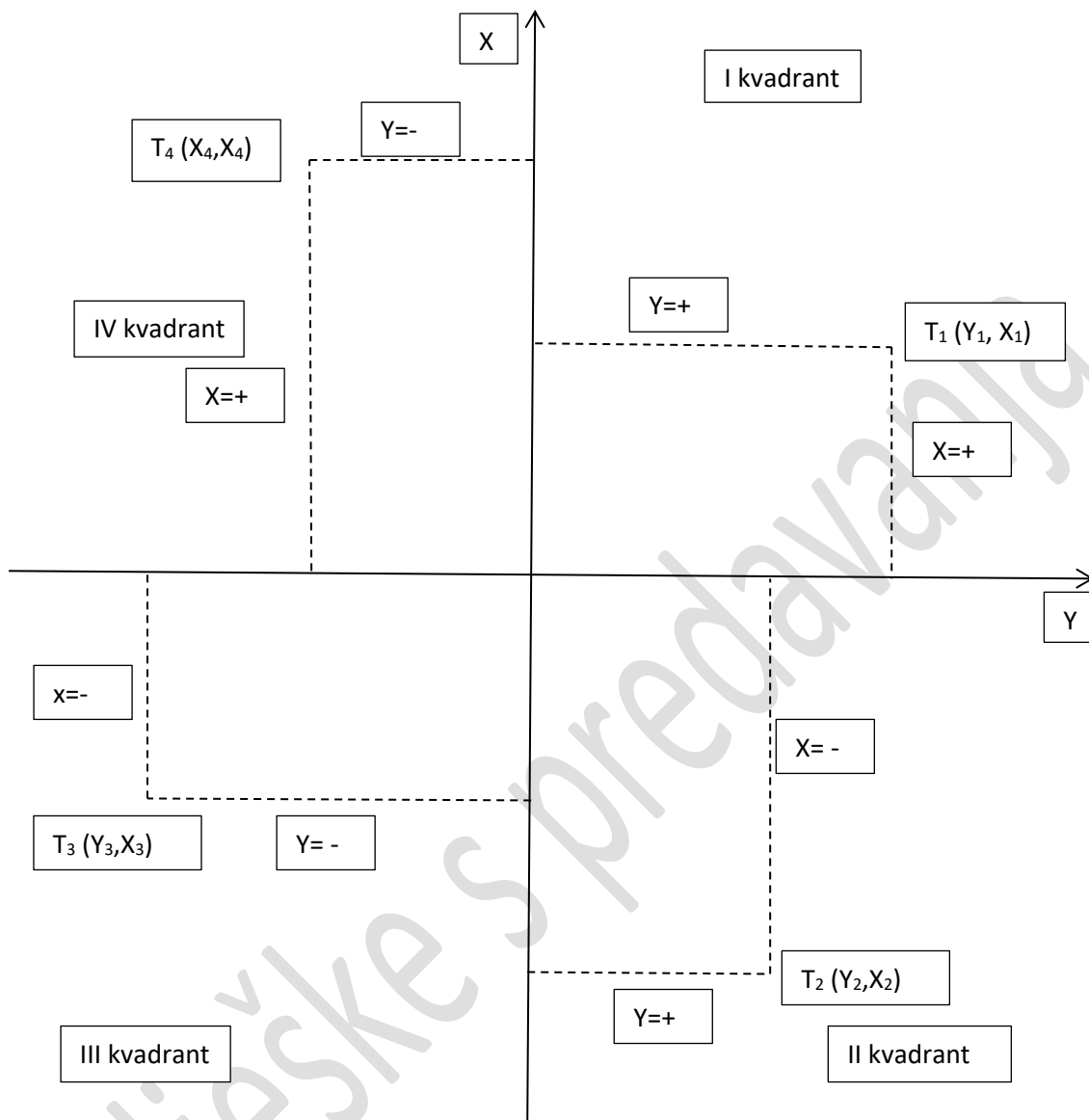


```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
utipkaj željeni broj:54
Broj je veći od nule.
>>> |
```



```
veći manji broj elif.pyw - C:\Python32\veći manji broj elif.pyw
File Edit Format Run Options Windows Help
#za uneseni broj želimo ustanoviti je li manji od nule,
#jednak nuli ili veći od nje.
#izbor jedne od tri mogućnosti s if-else strukturom
print("program će ispisati je li upisani broj manji,veći ili jednak nuli.")
broj=int(input("utipkaj željeni broj:"))
if broj==0:
    print("broj je jednak nuli.")
elif broj >0:
    print("broj je veći od nule")
else:
    print("broj je manji od nule.")
```

Bilješke s prečacima

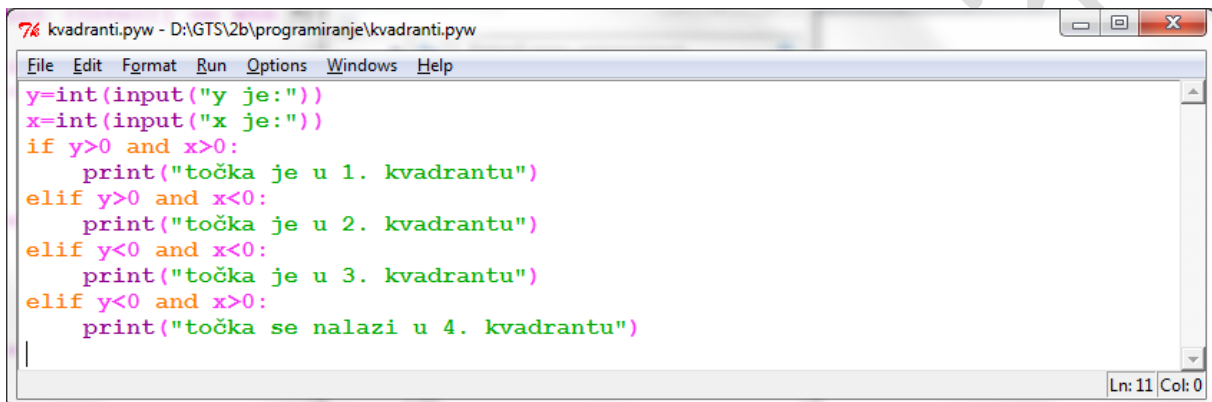


Ako je $y > 0$ and $x > 0$
print 1. Kvadrant

Ako je $y > 0$ and $x < 0$
print 2. Kvadrant

Ako je $y < 0$ and $x < 0$
print 3. Kvadrant

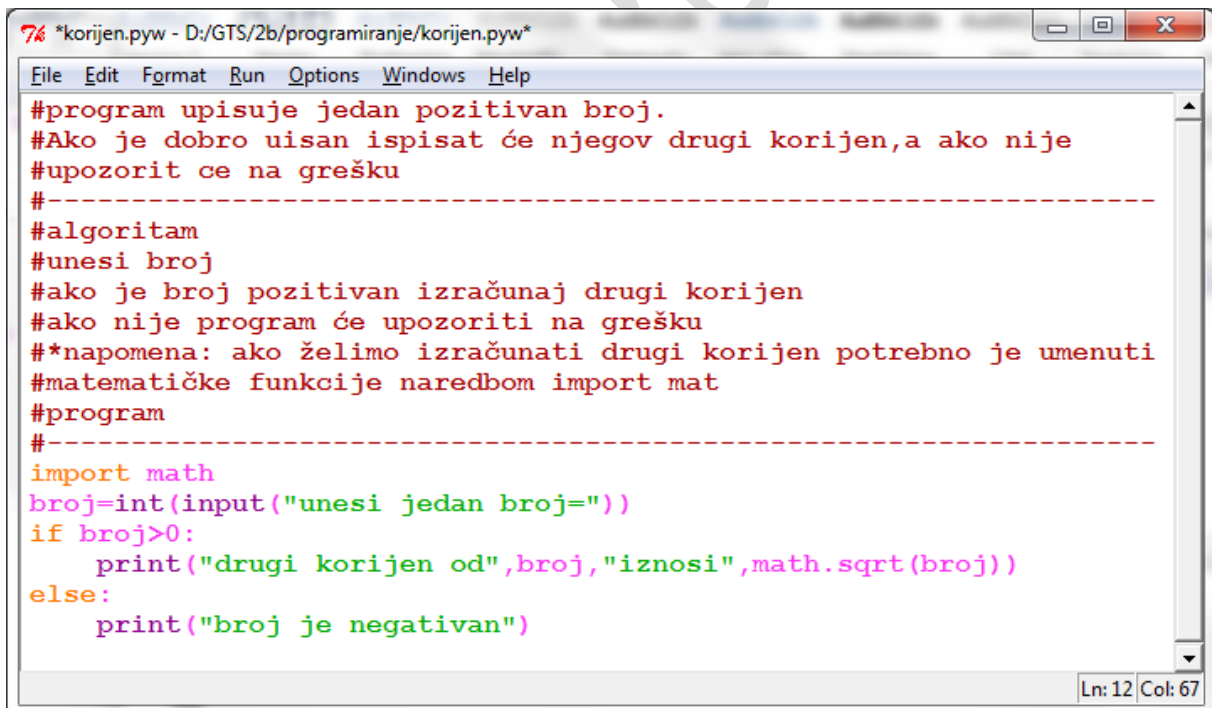
Ako je $y < 0$ and $x > 0$
print 4. Kvadrant



```

7% kvadranti.pyw - D:\GTS\2b\programiranje\kvadranti.pyw
File Edit Format Run Options Windows Help
y=int(input("y je:"))
x=int(input("x je:"))
if y>0 and x>0:
    print("točka je u 1. kvadrantu")
elif y>0 and x<0:
    print("točka je u 2. kvadrantu")
elif y<0 and x<0:
    print("točka je u 3. kvadrantu")
elif y<0 and x>0:
    print("točka se nalazi u 4. kvadrantu")
Ln: 11 Col: 0

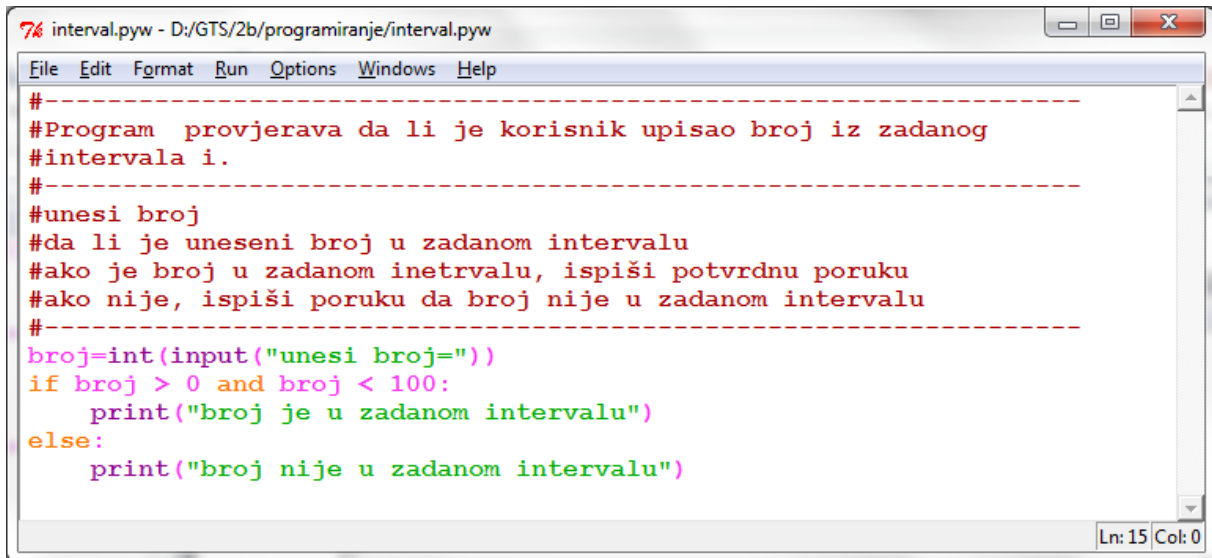
```



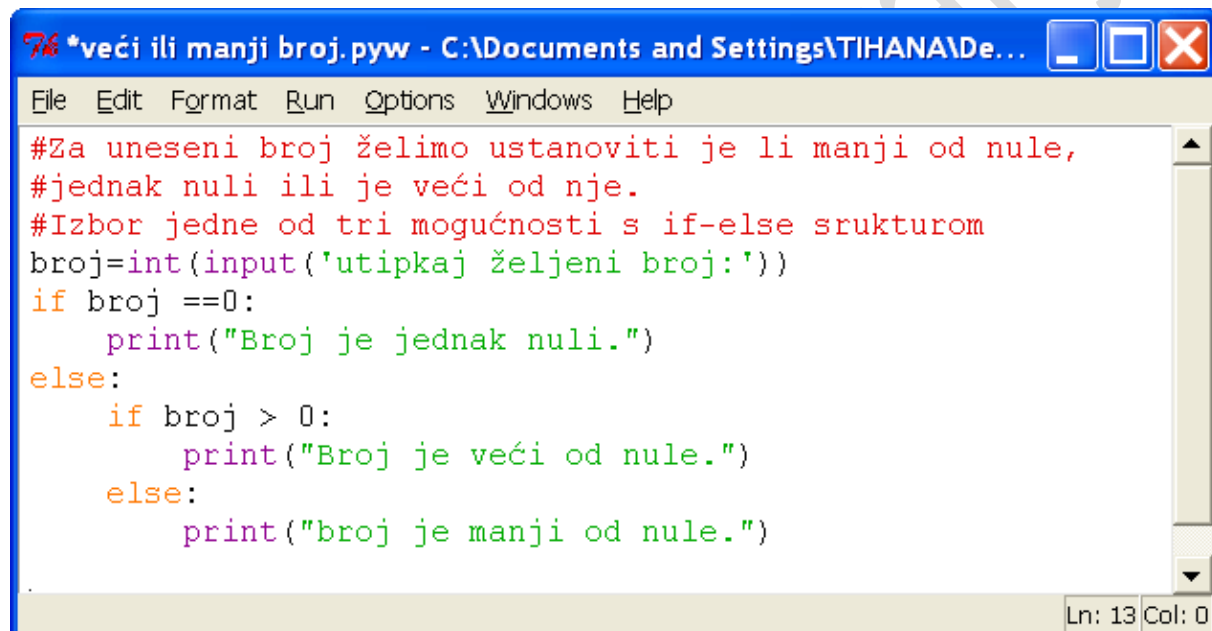
```

7% *korijen.pyw - D:\GTS\2b\programiranje/korijen.pyw*
File Edit Format Run Options Windows Help
#program upisuje jedan pozitivan broj.
#Ako je dobro uisan ispisat će njegov drugi korijen,a ako nije
#upozorit ce na grešku
#-----
#algoritam
#unesi broj
#ako je broj pozitivan izračunaj drugi korijen
#ako nije program će upozoriti na grešku
#*napomena: ako želimo izračunati drugi korijen potrebno je umenuti
#matematičke funkcije naredbom import mat
#program
#-----
import math
broj=int(input("unesi jedan broj="))
if broj>0:
    print("drugi korijen od",broj,"iznosi",math.sqrt(broj))
else:
    print("broj je negativan")
Ln: 12 Col: 67

```

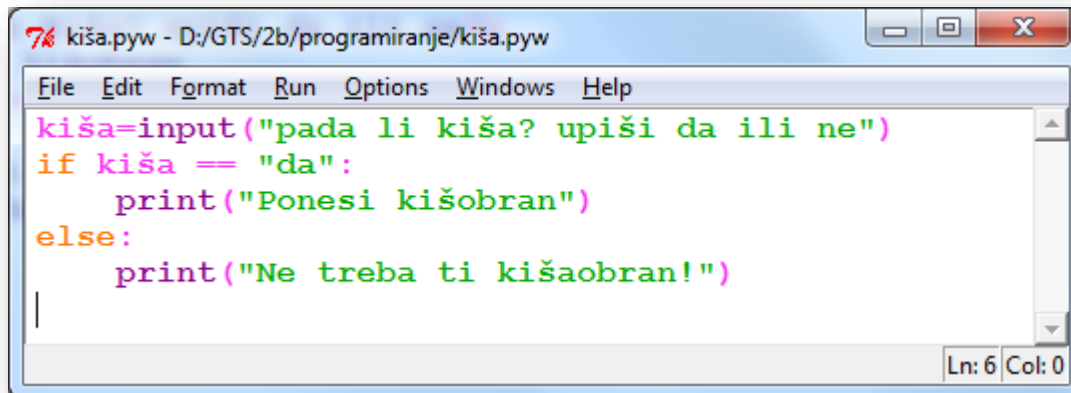


```
interval.pyw - D:/GTS/2b/programiranje/interval.pyw
File Edit Format Run Options Windows Help
#-----
#Program provjerava da li je korisnik upisao broj iz zadanog
#intervala i.
#-----
#unesi broj
#da li je uneseni broj u zadanom intervalu
#ako je broj u zadanom inetrvalu, ispiši potvrdnu poruku
#ako nije, ispiši poruku da broj nije u zadanom intervalu
#-----
broj=int(input("unesi broj="))
if broj > 0 and broj < 100:
    print("broj je u zadanom intervalu")
else:
    print("broj nije u zadanom intervalu")
Ln: 15 Col: 0
```

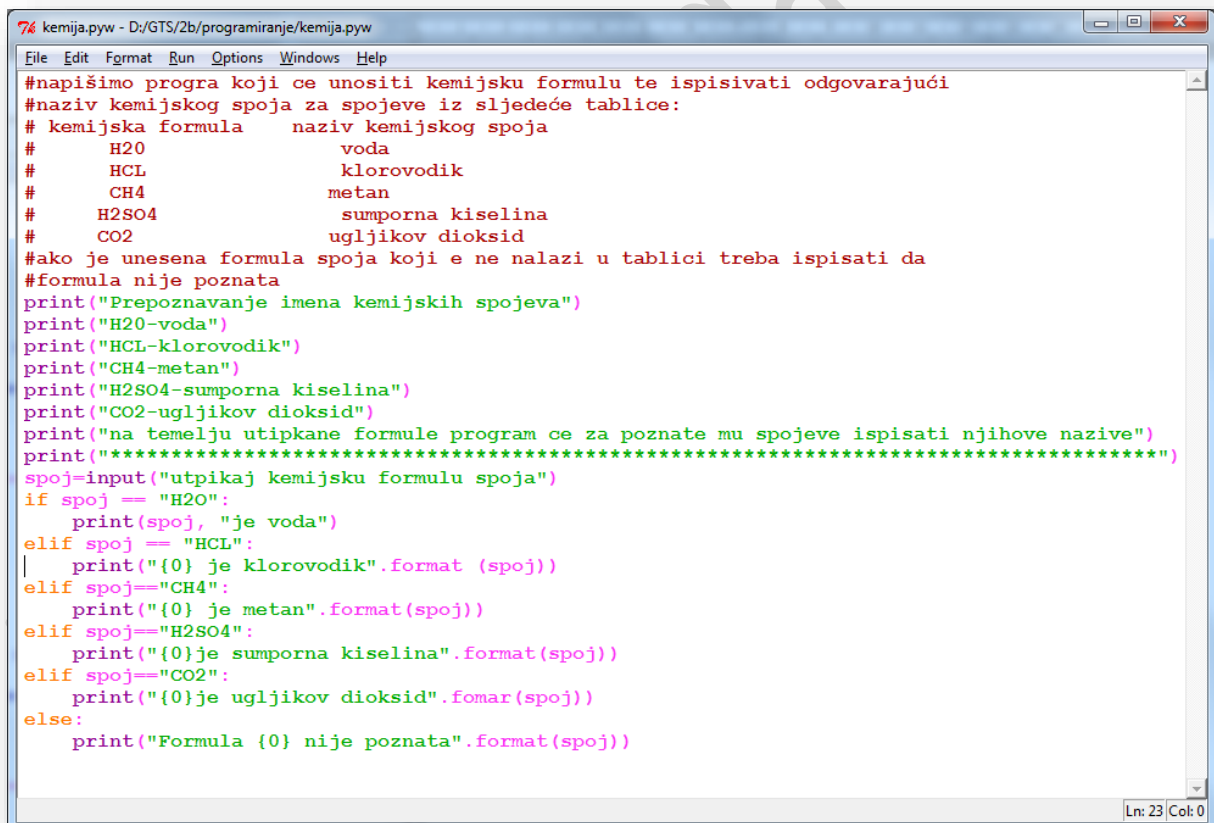


```
*veći ili manji broj.pyw - C:\Documents and Settings\TIHANA\De...
File Edit Format Run Options Windows Help
#Za uneseni broj želimo ustanoviti je li manji od nule,
#jednak nuli ili je veći od nje.
#Izbor jedne od tri mogućnosti s if-else strukturom
broj=int(input('utipkaj željeni broj:'))
if broj ==0:
    print("Broj je jednak nuli.")
else:
    if broj > 0:
        print("Broj je veći od nule.")
    else:
        print("broj je manji od nule.")
Ln: 13 Col: 0
```


NAKON 2. TESTA:



```
kiša.pyw - D:/GTS/2b/programiranje/kiša.pyw
File Edit Format Run Options Windows Help
kiša=input("pada li kiša? upiši da ili ne")
if kiša == "da":
    print("Ponesi kišobran")
else:
    print("Ne treba ti kišaobran!")
Ln: 6 Col: 0
```



```
kemija.pyw - D:/GTS/2b/programiranje/kemija.pyw
File Edit Format Run Options Windows Help
#napišimo progra koji ce unositi kemijsku formulu te ispisivati odgovarajući
#naziv kemijskog spoja za spojeve iz sljedeće tablice:
# kemijska formula    naziv kemijskog spoja
#   H2O                voda
#   HCL                klorovodik
#   CH4                metan
#   H2SO4             sumporna kiselina
#   CO2               ugljikov dioksid
#ako je unesena formula spoja koji e ne nalazi u tablici treba ispisati da
#formula nije poznata
print("Prepoznavanje imena kemijskih spojeva")
print("H2O-voda")
print("HCL-klorovodik")
print("CH4-metan")
print("H2SO4-sumporna kiselina")
print("CO2-ugljikov dioksid")
print("na temelju utipkane formule program ce za poznate mu spojeve ispisati njihove nazive")
print("*****")
spoj=input("utpikaj kemijsku formulu spoja")
if spoj == "H2O":
    print(spoj, "je voda")
elif spoj == "HCL":
    print("{0} je klorovodik".format(spoj))
elif spoj=="CH4":
    print("{0} je metan".format(spoj))
elif spoj=="H2SO4":
    print("{0}je sumporna kiselina".format(spoj))
elif spoj=="CO2":
    print("{0}je ugljikov dioksid".fomar(spoj))
else:
    print("Formula {0} nije poznata".format(spoj))
Ln: 23 Col: 0
```

```

*predmeti.pyw - C:\Documents and Settings\TIHANA\Desktop\Programiranje\predmeti.pyw*
File Edit Format Run Options Windows Help
print("Napiši program koji će ispisivati kad uneseš predmet tko ga predaje.")
print("matematika- prof. Portner")
print("povijest - prof. Bašić")
print("tjelesni - prof. Kantar")
print("fizika - prof. Šarić")
print("hrvatski - prof.Varović")
print("geodezija - prof. Jurković-Koren")
print("Kartografija - prof. Jurković- Koren")
print("fotogrametrija - prof. Tomasović")
print("programiranje - prof. Slaviček")
print("etika - prof. Bašić")
print("geodetska izmjera - prof. Jurković-Koren")
print("engleski - prof. Plemenčić")
print("nacrtna geometrija - prof. Žibrat")
print("OGI - prof. Šurina")
print("zemljopis - prof. Lovrec")
predmet=input("Unesi predmet=")
if predmet=="matematika":
    print(predmet, "U GTŠ predaje prof. Portner")
elif predmet=="povijest":
    print("{} u GTŠ predaje prof. Bašić".format(predmet))
elif predmet=="tjelesni":
    print("{} u GTŠ predaje prof. Kantar".format(predmet))
elif predmet=="fizika":
    print("{} u GTŠ predaje prof. Šarić".format(predmet))
elif predmet=="hrvatski":
    print("{} u GTŠ predaje prof. Varović".format(predmet))
elif predmet=="geodezija":
    print("{} u GTŠ predaje prof. Jurković- Koren".format(predmet))
elif predmet=="geodetska izmjera":
    print("{} u GTŠ predaje prof. Jurković- Koren".format(predmet))

```

Ln: 31 Col: 0

```

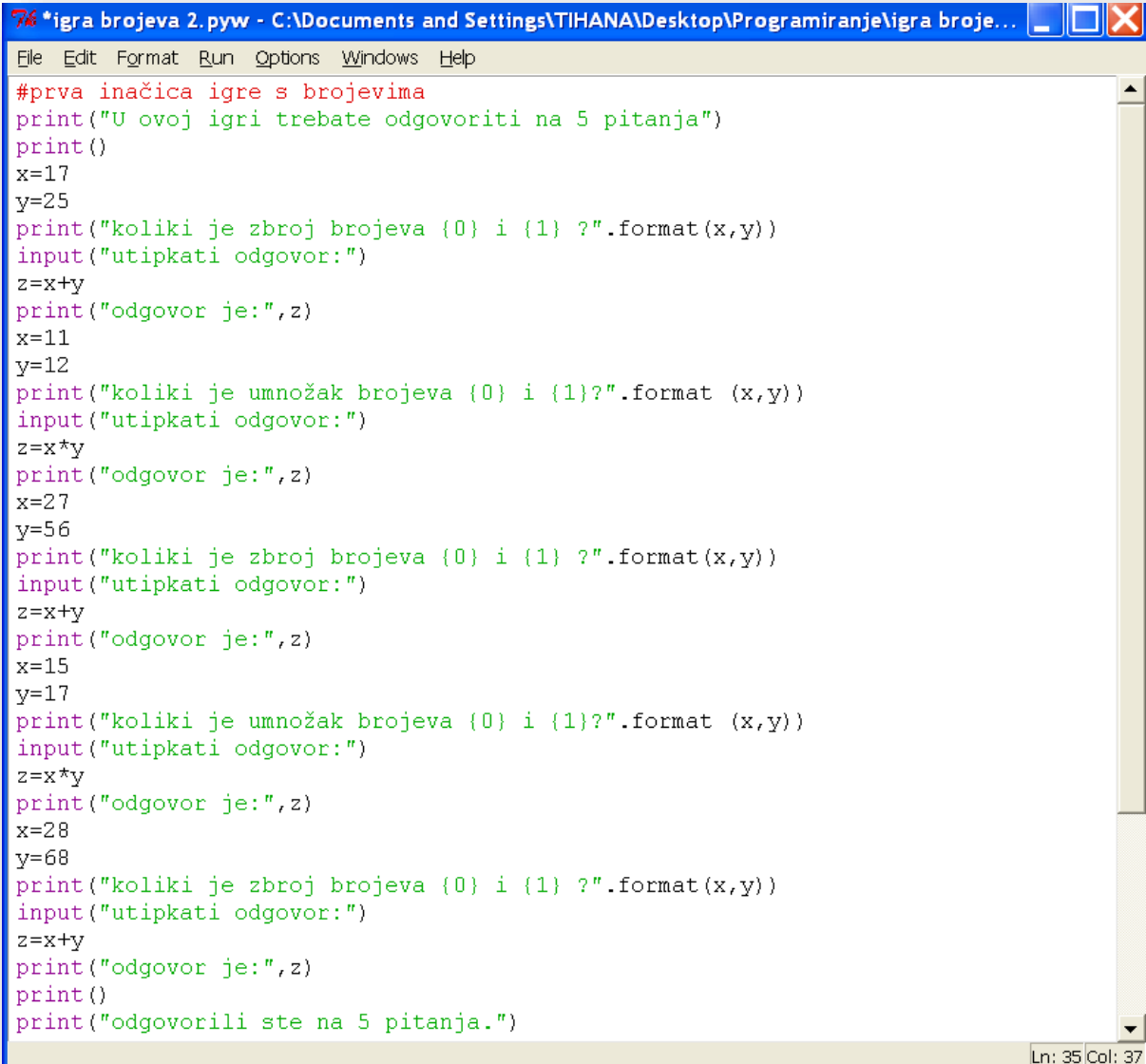
*predmeti.pyw - C:\Documents and Settings\TIHANA\Desktop\Programiranje\predmeti.pyw*
File Edit Format Run Options Windows Help
elif predmet=="kartografija":
    print("{} u GTŠ predaje prof. Jurković- Koren".format(predmet))
elif predmet=="fotogrametrija":
    print("{} u GTŠ predaje prof. Tomasović".format(predmet))
elif predmet=="programiranje":
    print("{} u GTŠ predaje prof. Slaviček".format(predmet))
elif predmet=="etika":
    print("{} u GTŠ predaje prof. Bašić".format(predmet))
elif predmet=="engleski":
    print("{} u GTŠ predaje prof. Plemenčić".format(predmet))
elif predmet=="nacrtna geometrija":
    print("{} u GTŠ predaje prof. Žibrat".format(predmet))
elif predmet=="OGI":
    print("{} u GTŠ predaje prof. Šurina".format(predmet))
elif predmet=="zemljopis":
    print("{} u GTŠ predaje prof. Lovrec".format(predmet))
else:
    print("tog predmeta nema u školi.")

```

Ln: 49 Col: 0

```
Otopine.pyw - C:\Documents and Settings\TIHANA\Desktop\Programiranje\Otopine.pyw
File Edit Format Run Options Windows Help
#kiselost i luznost otopina izrazavaju se pH vrijednostima. Napišimo program
#koji će utipkanu vrijednost pH vrijednosti napisati jeli otopina jako kisela,
#slabo kisela ili jako kisela lužina
#pri čemu su vrijednosti otopina dani sljedećom tablicom pH vrijednosti:
#<0,4.5) jako kisela
#<4.5, 6.5) slabo kisela
#<6.5 , 7.5) neutralna
#<7.5, 8.5) slabo lužnata
#<9.5, 14) jako lužnata
#klasifikacija otopina pH-vrijednosti
print("program ce ovisiti o utipkanj vrijednosti napisati jeli otopina jako, sl
ph=float(input("upisati pH vrijednost:"))
if ph<> 0 and ph<=4.5:
    print("otopina sa ph -vrijednoscu {0:0.1f} je jako kisela.".format(ph))
elif ph>4.5 and ph<=6.5:
    print("otopina sa ph -vrijednoscu {0:0.1f} je jako kisela.".format(ph))
elif ph>6.5 and ph<=7.5:
    print("otopina sa ph -vrijednoscu {0:0.1f} je jako kisela.".format(ph))
elif ph>7.5 and ph<=8.5:
    print("otopina sa ph -vrijednoscu {0:0.1f} je jako kisela.".format(ph))
elif ph>9.5 and ph<=14:
    print("otopina sa ph -vrijednoscu {0:0.1f} je jako kisela.".format(ph))
else:
    print|
Ln: 24 Col: 9
```

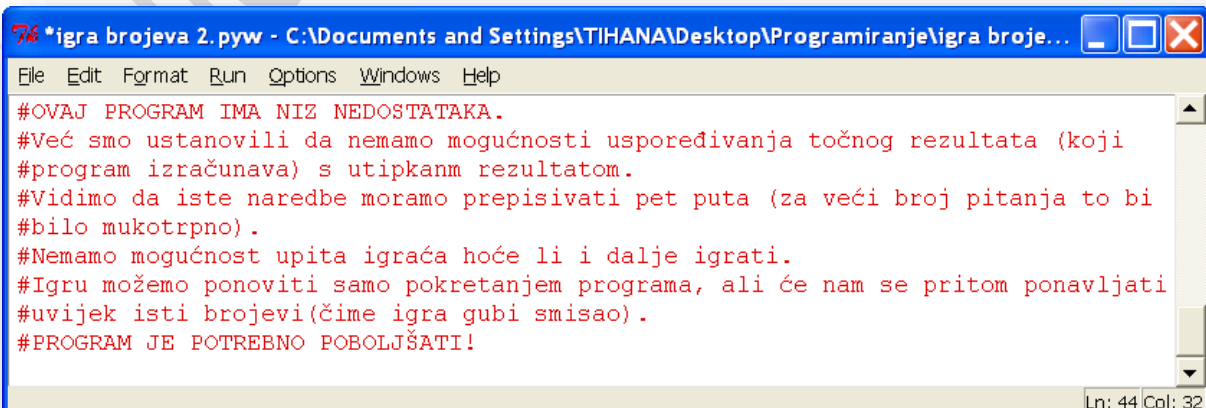
Bilješke s pri

Zamišljena igra brojevima (2):


```

*igra brojeva 2. pyw - C:\Documents and Settings\TIHANA\Desktop\Programiranje\igra broje...
File Edit Format Run Options Windows Help
#prva inačica igre s brojevima
print("U ovoj igri trebate odgovoriti na 5 pitanja")
print()
x=17
y=25
print("koliki je zbroj brojeva {0} i {1} ?".format(x,y))
input("utipkati odgovor:")
z=x+y
print("odgovor je:",z)
x=11
y=12
print("koliki je umnožak brojeva {0} i {1}?".format (x,y))
input("utipkati odgovor:")
z=x*y
print("odgovor je:",z)
x=27
y=56
print("koliki je zbroj brojeva {0} i {1} ?".format(x,y))
input("utipkati odgovor:")
z=x+y
print("odgovor je:",z)
x=15
y=17
print("koliki je umnožak brojeva {0} i {1}?".format (x,y))
input("utipkati odgovor:")
z=x*y
print("odgovor je:",z)
x=28
y=68
print("koliki je zbroj brojeva {0} i {1} ?".format(x,y))
input("utipkati odgovor:")
z=x+y
print("odgovor je:",z)
print()
print("odgovorili ste na 5 pitanja.")
Ln: 35 Col: 37

```



```

*igra brojeva 2. pyw - C:\Documents and Settings\TIHANA\Desktop\Programiranje\igra broje...
File Edit Format Run Options Windows Help
#OVAJ PROGRAM IMA NIZ NEDOSTATAKA.
#Već smo ustanovili da nemamo mogućnosti uspoređivanja točnog rezultata (koji
#program izračunava) s utipkanm rezultatom.
#Vidimo da iste naredbe moramo prepisivati pet puta (za veći broj pitanja to bi
#bilo mukotrпно).
#Nemamo mogućnost upita igrača hoće li i dalje igrati.
#Igru možemo ponoviti samo pokretanjem programa, ali će nam se pritom ponavljati
#uvijek isti brojevi(čime igra gubi smisao).
#PROGRAM JE POTREBNO POBOLJŠATI!
Ln: 44 Col: 32

```

```
76 *geometrijska tijela.pyw - C:\Documents and Settings\TIHANA\Desкто...
File Edit Format Run Options Windows Help
print("Izračunavanje opsega i volumena određenog lika." )
print("kockaV - a*a*a")
print("kockaO - 6*a*a")
print("kvadarV - a*b*c")
print("kvadarO - 2*(a*b+a*c+b*c) ")
print("valjakV - r*r*pi*v")
print("valjakO - 2*r*pi*(r+v) ")
tijelo=input("Unesi tijelo te uz njega V ili O=")
if tijelo=="kockaV":
    a=int(input("Unesi stranicu a="))
    print("Volumen kocke iznosi",a*a*a)
elif tijelo=="kockaO":
    a=int(input("Unesi stranicu a"))
    print("Opseg kocke iznosi", 6*a*a)
elif tijelo=="kvadarV":
    a=int(input("unesi stranicu a="))
    b=int(input("unesi stranicu b="))
    c=int(input("unesi stranicu c="))
    print("Volumen kvadra iznosi",a*b*c)
elif tijelo=="kvadarO":
    a=int(input("unesi stranicu a="))
    b=int(input("unesi stranicu b="))
    c=int(input("unesi stranicu c="))
    print("Opseg kvadra iznosi",2*(a*b+a*c+b*c))
elif tijelo=="valjakV":
    pi=3.14
    r=int(input("unesi radijus="))
    v=int(input("unesi visinu="))
    print("volumen valjka iznosi",r*r*pi*v)
elif tijelo=="valjakO":
    pi=3.14
    r=int(input("unesi radijus="))
    v=int(input("unesi visinu="))
    print("opseg valjka iznosi",2*r*pi*(r+v))|
Ln: 34 Col: 45
```

```
7% Pitanja a,b,c,d.pyw - D:\GTS\2b\programiranje\Pitanja a,b,c,d.pyw
File Edit Format Run Options Windows Help
#pitanja a,b,c,d
#
#Broj_točnih_odgovora
broj_pitanja=8
print("Trebate odgovoriti na {0} pitanja \n".format(broj_pi
broj_točnih_odgovora=0
print("1.Koja je najpoznatija društvena mreža?")
print("a) MSN")
print("b) Facebook")
print("c) Twitter")
print("d) Account")
odgovor=input("Unesi točan odgovor: a,b,c ili d=")
if odgovor=="b":
    print("Točno")
    broj_točnih_odgovora+=1
else:
    print ("Netočno")
    print("Točan odgovor je b) Facebook")
#*****
print()
print()
print("2.Kada je započeo razvoj Pythona?")
print("a) 1991. god")
print("b) 2007. god")
print("c) 1987. god")
print("d) 1992. god")
odgovor=input("Unesi točan odgovor: a,b,c ili d=")
if odgovor=="a":
    print("Točno")
    broj_točnih_odgovora+=1
else:
    print ("Netočno")
    print("Točan odgovor je a) 1991. god")
#*****
print()
print()
print("3.Pravokutnik u dijagramu tijeka označava?")
print("a) početak")
print("b) ulaz podataka")
print("c) odluka")
print("d) naredba")
odgovor=input("Unesi točan odgovor: a,b,c ili d=")
if odgovor=="d":
    print("Točno")
    broj_točnih_odgovora+=1
else:
    print ("Netočno")
    print("Točan odgovor je d) naredba")
Ln: 2 Col: 1
```

```
7 Pitanja a,b,c,d.pyw - D:\GTS\2b\programiranje\Pitanja a,b,c,d.pyw
File Edit Format Run Options Windows Help
#####
print()
print()
print("4. Koja je prva generacija programskog jezika?")
print("a) strojni jezik")
print("b) simbolički (asemblerski) jezici")
print("c) jezici prilagođeni krajnjim korisnicima")
print("d) jezici za programiranje visoke razine")
odgovor=input("Unesi točan odgovor: a,b,c ili d=")
if odgovor=="a":
    print("Točno")
    broj_točnih_odgovora+=1
else:
    print ("Netočno")
    print("Točan odgovor je a) strojni jezik")
#####
print()
print()
print("5.Koji program ne pripada četvrtoj generaciji progra
print("a) SQL")
print("b) GIS")
print("c) LOGO")
print("d) System")
odgovor=input("Unesi točan odgovor: a,b,c ili d=")
if odgovor=="c":
    print("Točno")
    broj_točnih_odgovora+=1
else:
    print ("Netočno")
    print("Točan odgovor je c) LOGO")
#####
print()
print()
print("6.Tko je izradio bušenu katicu 1880. godine?")
print("a) Herman Hollerith")
print("b) Charles Babbage")
print("c) Joseph Marie Charles Jacquard ")
print("d) Ada Lovelace Byron")
odgovor=input("Unesi točan odgovor: a,b,c ili d=")
if odgovor=="a":
    print("Točno")
    broj_točnih_odgovora+=1
else:
    print ("Netočno")
    print("Točan odgovor je a) Herman Hollerith")
#####
print()
print()
```

```

Pitanja a,b,c,d.pyw - D:\GTS\2b\programiranje\Pitanja a,b,c,d.pyw
File Edit Format Run Options Windows Help
print()
print("7.Tko je bio prvi programer/ka?")
print("a) Herman Hollerith")
print("b) Ada Lovelace Byron")
print("c) Joseph Marie Charles Jacquard")
print("d) Charles Babbage")
odgovor=input("Unesi točan odgovor: a,b,c ili d=")
if odgovor=="b":
    print("Točno")
    broj_točnih_odgovora+=1
else:
    print ("Netočno")
    print("Točan odgovor je b) Ada Lovelace Byron")
#*****
print()
print()
print("8.Tko je razvio Python?")
print("a) Herman Hollerith")
print("b) Charles Babbage")
print("c) Guido Van Rossum ")
print("d) Ada Lovelace Byron")
odgovor=input("Unesi točan odgovor: a,b,c ili d=")
if odgovor=="c":
    print("Točno")
    broj_točnih_odgovora+=1
else:
    print ("Netočno")
    print("Točan odgovor je c) Guido Van Rossum")
#*****
Ln: 2 Col: 1

```

```

Pitanja a,b,c,d.pyw - D:\GTS\2b\programiranje\Pitanja a,b,c,d.pyw
File Edit Format Run Options Windows Help
#*****
print()
print()
print("\nOdgovorili ste na {} pitanja. \n".format(broj_pitanja))
print("Od toga je {} točnih.".format(broj_točnih_odgovora))
postotak=100*broj_točnih_odgovora / broj_pitanja
print("To znači da imate {} posto točnih odgovora. \n".format(postotak))
print("Za završetak programa pritisnite tipku <enter> - lijep pozdrav!")
Ln: 2 Col: 1

```



```
Python Shell
File Edit Shell Debug Options Windows Help
>>> x=25
>>> import math
>>> y=math.sqrt (x)
>>> y
5.0
>>> x=81
>>> y=math.sqrt (x)
>>> y
9.0
>>> #####
>>> from math import sqrt
>>> x=101
>>> y=math.sqrt (x)
>>> y
10.04987562112089
>>> print(sqrt(72))
8.48528137423857
>>> print (sqrt(169))
13.0
>>> #####
>>> #učitavanje svih funkcija iz modula pomoću znaka "*"
>>> from math import *
>>> print(sqrt(2))
1.4142135623730951
>>> x=5
>>> y=7
>>> z=x*x+y*y
>>> print(sqrt(z))
8.602325267042627
>>> x=25
>>> y=37
>>> z=x**2+y**2
>>> print(sqrt(z))
44.654227123532216
Ln: 44 Col: 4
```

```
Python Shell
File Edit Shell Debug Options Windows Help
>>> pi
3.141592653589793
>>> r=7
>>> print("Površina kruga s polumjerom {0} iznosi {1}.".format(r, pi*r**2))
Površina kruga s polumjerom 7 iznosi 153.93804002589985.
>>>
Ln: 52 Col: 4
```

Moduli

Zbog toga je preporučljivo za uvoz modula koristiti se naredbom:

import ime_modula

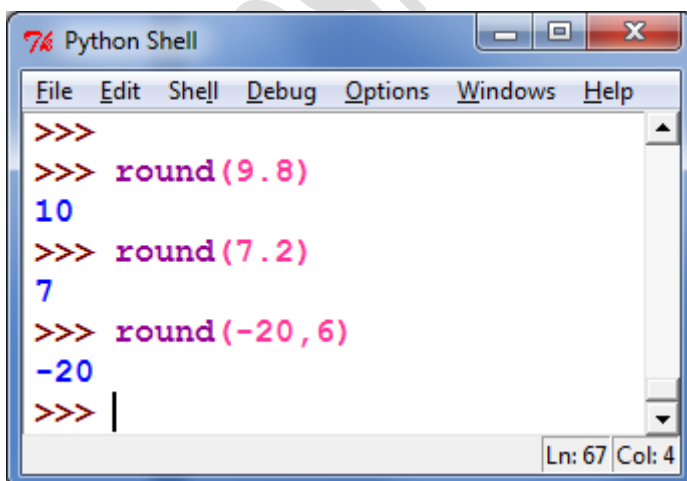
i zatim ispred imena svake funkcije u prefiksu napisati ime_modula ili uz uvoz modula naznačiti funkcije kojima ćemo se koristiti:

from ime_modula import ime_funkcije_1, ime_funkcije_2,...

S obzirom na to da ćemo funkcije modula math često upotrebljavati, u tablici 5.7. nalazi se popis najčešće korištenih funkcija ovog modula:

Funkcija	Opis	Primjer
ceil(x)	najmanji cijeli broj veći od ili jednak x	ceil(3.4)=4 ceil(4.9)=5
floor(x)	najveći cijeli broj manji od ili jednak x	floor(3.2)=3 floor(7.8)=7
exp(x)	e^x	exp(1)=2.71828
log(x,b)	$\log_b x$	log(100,10)=2.0
sin(x)	sin x	sin(pi/2)=1.0
cos(x)	cos x	cos(pi)=1.0
tan(x)	tg x	tan(0)
asin(x)	Arcus sinusa	asin(1)=1.57079
acos(x)	Arcus cosinus	acos(-1)=3.14159

Funkcija **round(x)** vraća cijeli broj koji je najbliži realnom broju.



```

Python Shell
File Edit Shell Debug Options Windows Help
>>>
>>> round(9.8)
10
>>> round(7.2)
7
>>> round(-20,6)
-20
>>> |
Ln: 67 Col: 4

```

Modul random

Za generiranje brojeva koristit ćemo se funkcijama iz modula koji se zove **random**. Modul **random** sadržava funkciju kojima se može generirati nasumične brojeve i modelirati slučajne događaje i pojave.

U modulu **random** postoji funkcija **randint(a,b)** kojom možemo oponašati različite primjere zadataka. Svaki put kad se ta funkcija pozove ona će vratiti jedan cijeli broj iz zatvorenog intervala (a,b) (tj. i brojeve koji su jednaki granicama a i b).

Prema tome, funkcija **randint(0,1)** može oponašati novčić (ako glavu označimo brojem 0 i pismo brojem 1), funkcija **randint(1,6)** može oponašati kocku, a funkcija **randint(1,N)** snop karata.

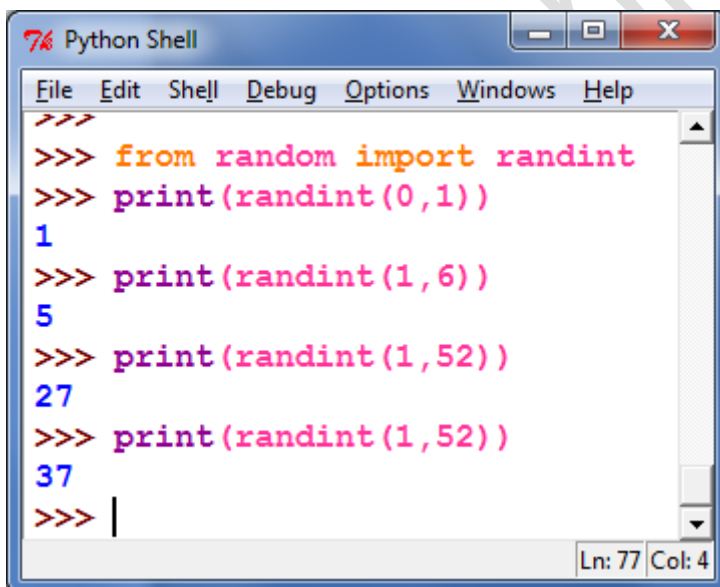
Funkcija **randit(a,b)** ne određuje ishode sasvim slučajno- ona ih izračunava nekim algoritmom koji jako dobro oponaša slučajnost.

Zbog toga se brojevi koje ona vraća mogu nazvati *pseudoslučajnim*.

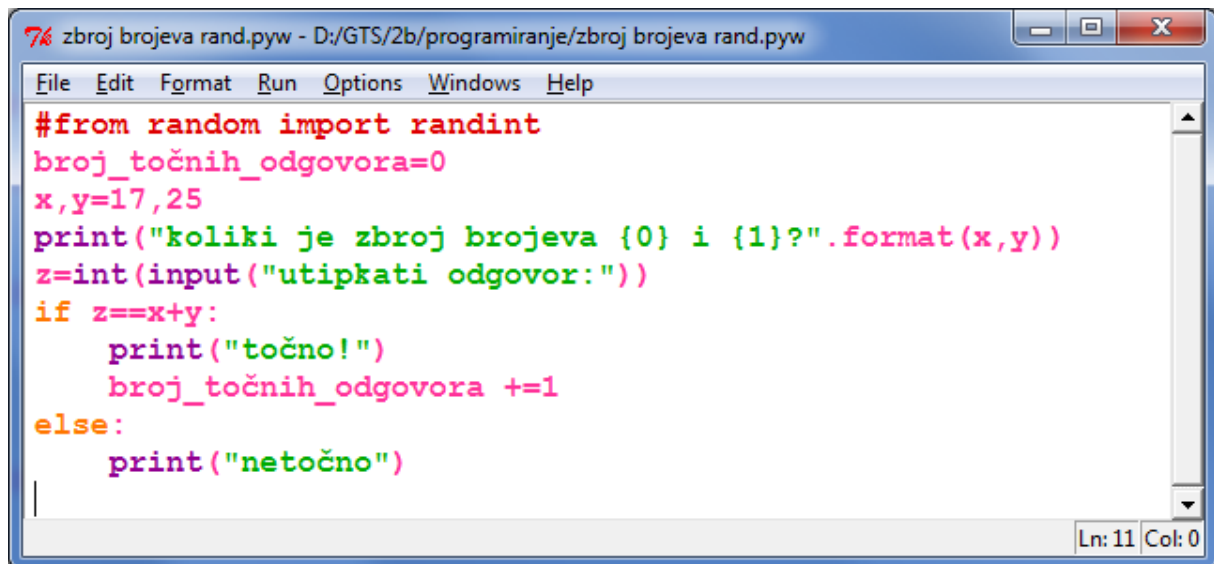
Mi ćemo ih nazvati *nasumično odabranim brojevima* ili kraće samo *nasumičnim brojevima*.

Funkcija **randint(a,b)** može nam poslužiti kao generator nasumičnih brojeva.

Pogledajmo brojeve koje dobivamo u interaktivnom sučelju:



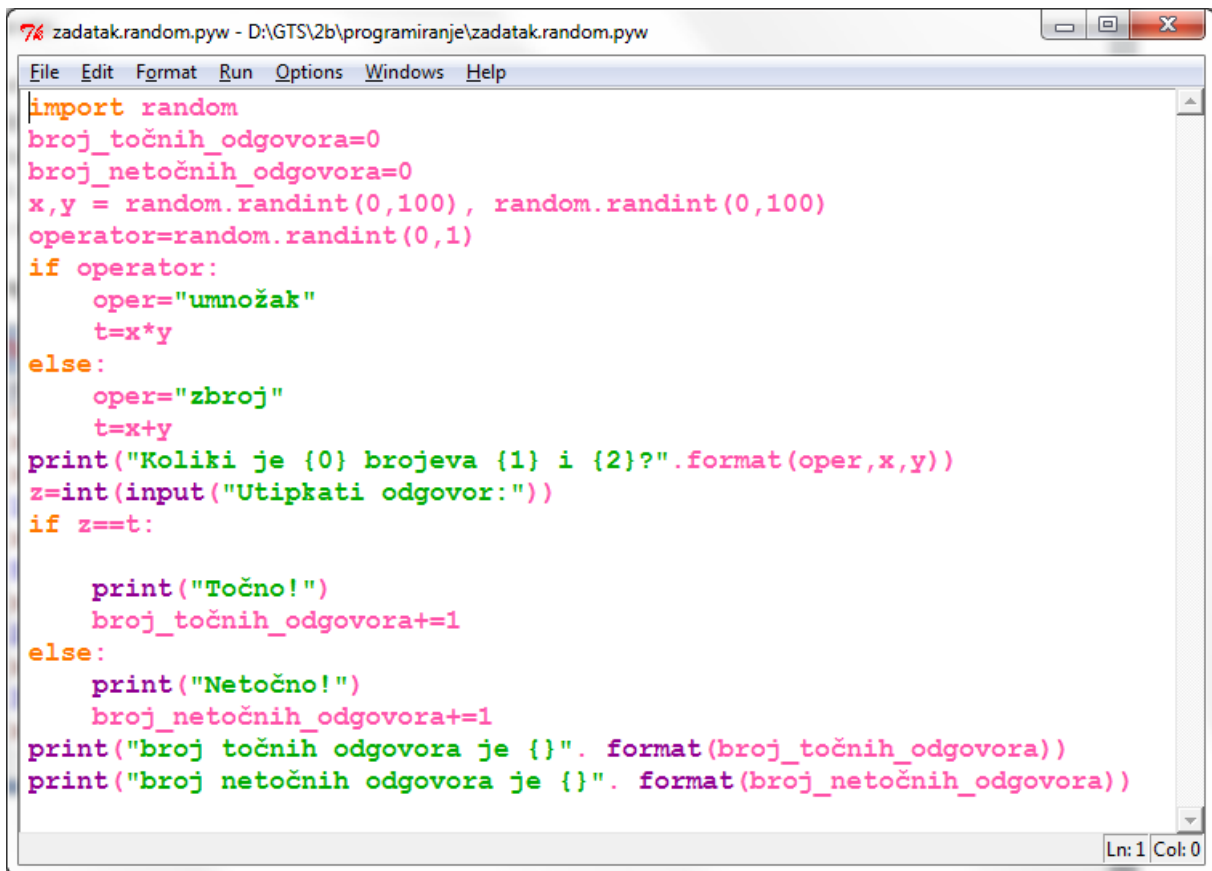
```
Python Shell
File Edit Shell Debug Options Windows Help
>>>
>>> from random import randint
>>> print(randint(0,1))
1
>>> print(randint(1,6))
5
>>> print(randint(1,52))
27
>>> print(randint(1,52))
37
>>> |
Ln: 77 Col: 4
```



```
7% zbroj brojeva rand.pyw - D:/GTS/2b/programiranje/zbroj brojeva rand.pyw
File Edit Format Run Options Windows Help
#from random import randint
broj_točnih_odgovora=0
x,y=17,25
print("koliki je zbroj brojeva {0} i {1}?".format(x,y))
z=int(input("utipkati odgovor:"))
if z==x+y:
    print("točno!")
    broj_točnih_odgovora +=1
else:
    print("netočno")
Ln: 11 Col: 0
```

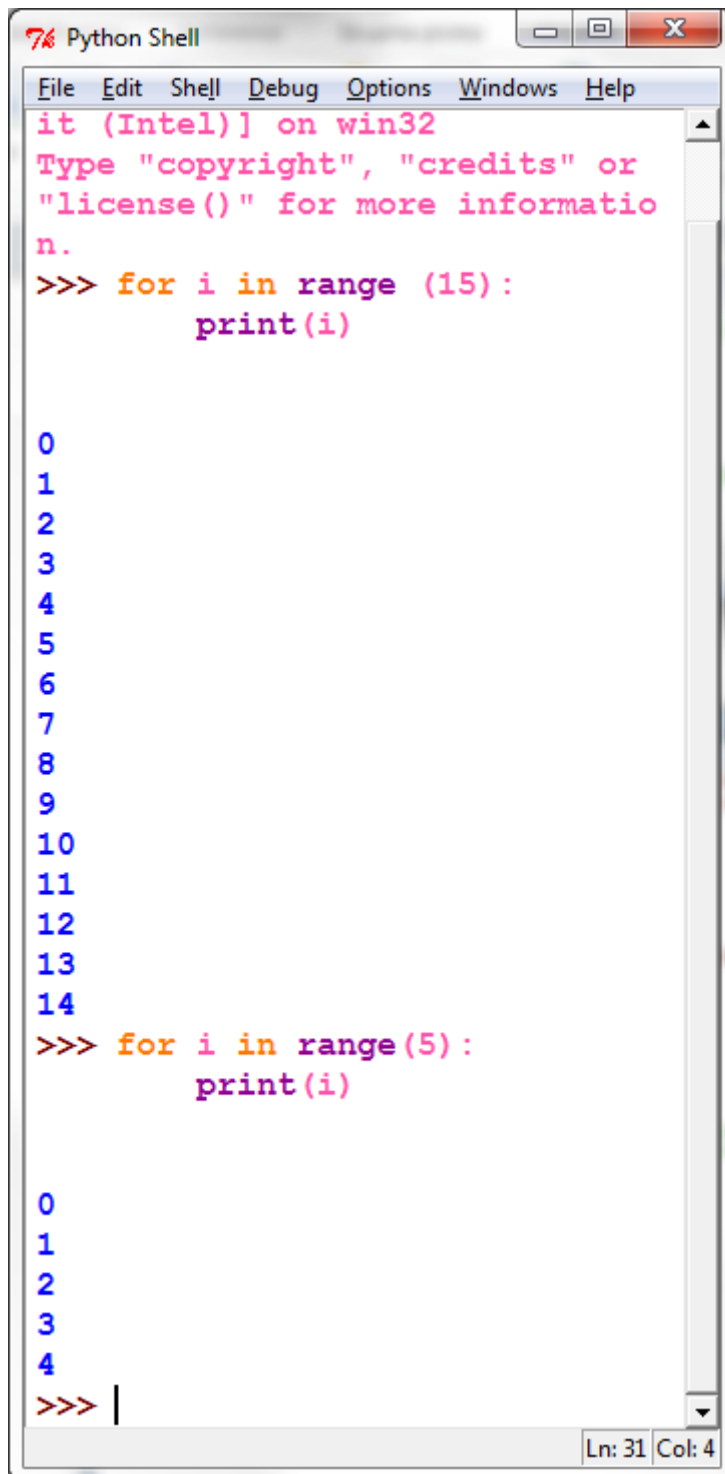
Bilješke s predavanja

POSLIJE TREĆEG TESTA



```
zadatak.random.pyw - D:\GTS\2b\programiranje\zadatak.random.pyw
File Edit Format Run Options Windows Help
import random
broj_točnih_odgovora=0
broj_netočnih_odgovora=0
x,y = random.randint(0,100), random.randint(0,100)
operator=random.randint(0,1)
if operator:
    oper="umnožak"
    t=x*y
else:
    oper="zbroj"
    t=x+y
print("Koliki je {0} brojeva {1} i {2}?" .format(oper,x,y))
z=int(input("Utipkati odgovor:"))
if z==t:
    print("Točno!")
    broj_točnih_odgovora+=1
else:
    print("Netočno!")
    broj_netočnih_odgovora+=1
print("broj točnih odgovora je {}".format(broj_točnih_odgovora))
print("broj netočnih odgovora je {}".format(broj_netočnih_odgovora))
Ln: 1 Col: 0
```

Bilješke



```
Python Shell
File Edit Shell Debug Options Windows Help
it (Intel) on win32
Type "copyright", "credits" or
"license()" for more informatio
n.
>>> for i in range (15):
    print(i)

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
>>> for i in range (5):
    print(i)

0
1
2
3
4
>>> |
```

Ln: 31 Col: 4

```
Python Shell
File Edit Shell Debug Options Windows Help
>>> for i in range(20):
    print(i, sep = " ")

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
>>> |
Ln: 55 Col: 4
```

```
Python Shell
File Edit Shell Debug Options Windows Help
>>> for i in range(5):
    print(i, end=" ")

0 1 2 3 4
>>> for i in range(5):
    print(i, end=" ")

0          1          2          3          4
>>>
Ln: 65 Col: 4
```

```
Python Shell
File Edit Shell Debug Options Windows Help
>>> for i in range(5):
    print(i, end="-")

0-1-2-3-4-
>>> for i in range(5):
    print(i, end=" - ")

0 - 1 - 2 - 3 - 4 -
>>> |
Ln: 75 Col: 4
```

```
Python Shell
File Edit Shell Debug Options Windows Help
>>> for i in range (5):
        print(i,end=" # ")

0 # 1 # 2 # 3 # 4 #
>>> |
Ln: 80 Col: 4
```

```
Python Shell
File Edit Shell Debug Options Windows Help
>>> for i in range (5):
        print(i,end=" Renata ")

0 Renata 1 Renata 2 Renata 3 Renata 4 Renata
>>> |
Ln: 85 Col: 4
```

```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.11.0 (tags v.11.0.0 2022-10-10) on win32
Type "copyright", "credits" or "license()" for more information.
>>> for i in range(100):
        print(i,end="")
        if not ((i+1) % 10):
            print()

0123456789
10111213141516171819
20212223242526272829
30313233343536373839
40414243444546474849
50515253545556575859
60616263646566676869
70717273747576777879
80818283848586878889
90919293949596979899
>>>
Ln: 19 Col: 4
```



```

Python Shell
File Edit Shell Debug Options Windows Help
>>> for i in range(100):
        print(" {:3d}".format(i),end="")
        if not ((i+1) % 10):
            print()

    0  1  2  3  4  5  6  7  8  9
   10 11 12 13 14 15 16 17 18 19
   20 21 22 23 24 25 26 27 28 29
   30 31 32 33 34 35 36 37 38 39
   40 41 42 43 44 45 46 47 48 49
   50 51 52 53 54 55 56 57 58 59
   60 61 62 63 64 65 66 67 68 69
   70 71 72 73 74 75 76 77 78 79
   80 81 82 83 84 85 86 87 88 89
   90 91 92 93 94 95 96 97 98 99
>>> |
Ln: 46 Col: 4

```

Zadatak 1.

Napišimo program za ocjenu generatora nasumičnih brojeva. Program treba simulirati bacanje novčića n puta te treba ispisati vjerojatnost pojavljivanja pisma odnosno glave.

```

*bacanje novčića.pyw - D:/GTS/2b/programiranje/bacanje novčića.pyw*
File Edit Format Run Options Windows Help
#ispitivanje modela bacanja novčića
from random import randint
print("Program modelira bacanje novčića s N bacanja i ispisuje dobivene"
      "vjerojatnosti pojavljivanja pisma i glave")
N=int(input("Upiši željeni broj bacanja novčića N:"))
broj_pisma=0
broj_glava=0
for i in range (N):
    bacanje=randint(0,1)
    if bacanje==0:
        broj_pisma +=1
    else:
        broj_glava +=1
vjerojatnost_pisma=broj_pisma/N
vjerojatnost_glave=broj_glava / N
print("Za {0} bacanja novčića utvrđene su sljedeće vjerojatnosti:\
nvjerojatnost_pisma={1:.2f}nvjerojatnost_glave={2:.2f}"
      .format(N,vjerojatnost_pisma,vjerojatnost_glave))
Ln: 18 Col: 0

```

```

7% visina-while.pyw - D:/GTS/2b/programiranje/visina-while.pyw
File Edit Format Run Options Windows Help
#Izračunavanje srednje visine petljom while
print("Izračunaj srednje visine proizvoljnog broja osoba")
brojilo=0
zbroj=0.0
imabrojeva="da"
while imabrojeva=="da":
    broj=float(input("Upisati visinu u m:"))
    zbroj+= broj
    brojilo+=1
    imabrojeva=input("Ima li jos unosa(utipkaj da ili ne)?")
print('\nSrednja visina {} osoba u metrima{:.2f}'.format(brojilo, zbroj/brojilo)
Ln: 11 Col: 0

```

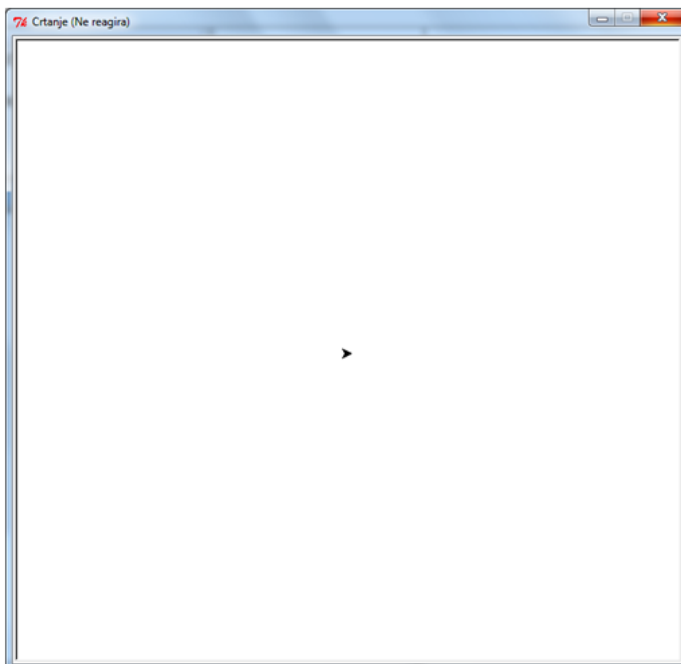
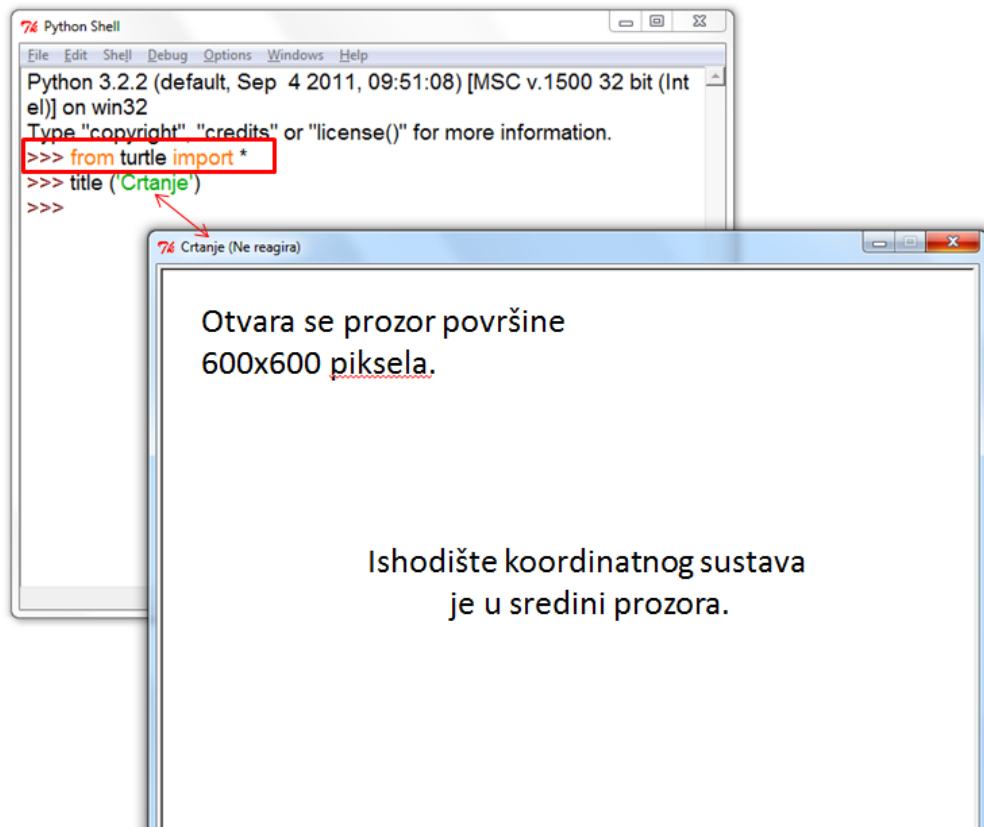
```

7% igra brojevima4.pyw - D:/GTS/2b/programiranje/igra brojevima4.pyw
File Edit Format Run Options Windows Help
#Igra s brojevima (četvrta inačica)
import random
random.seed()
nastavak="da"
while nastavak == "da":
    brojpitjanja=random.randint(2,5)
    print("U ovoj igri trebate odgovoriti na {} pitanja!\n".format(brojpitjanja))
    brojtočnihodgovora=0
    for i in range (brojpitjanja):
        x,y=random.randint(10,20), random.randint(10,20)#interval operanada
        operator=random.randint(0,1)
        if operator:
            oper="umnožak"
            t=x*y
        else:
            oper="zbroj"
            t=x+y
        print("Koliki je {0} brojeva {1} i {2}?".format(oper,x,y))
        z=int(input("Utipkati odgovor:"))
        if z==t:
            print("Točno!")
            brojtočnihodgovora+=1
            print("-----")
        else:
            print("Netočno!")
            print("-----")
    print("\nOdgovorili ste na {} pitanja. \n".format(brojpitjanja))
    print("Od toga je {} točnih.".format(brojtočnihodgovora))
    postotak=100*brojtočnihodgovora/brojpitjanja
    print("To znači da imaš {}% točnih odgovora.\n".format(postotak))
    nastavak=input("Želite li nastaviti igru (da ili ne)?")
input("Za završetak programa pritisnite tipku <Enter> - Lijep pozdrav!")
Ln: 27 Col: 40

```

Osnove računalne grafike

Grafički modul



```
>>> reset()
```

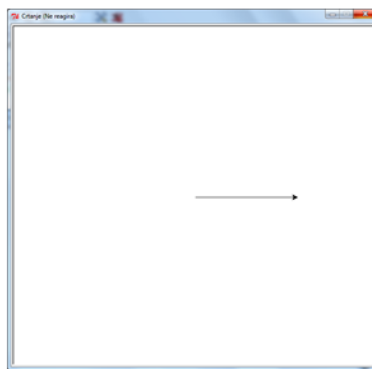
- Pojavit će se u ishodištu pero (strelica glave kornjače) usmjerena u smjeru pozitivne osi x.

Osnovne funkcije za gibanje pera

Funkcija	Alternativni naziv	Opis djelovanja funkcije
<code>forward(d)</code>	<code>fd(d)</code>	Pomiče pero za d jedinica naprijed
<code>backward(d)</code>	<code>back(d)</code> , <code>bk(d)</code>	Pomiče pero za d jedinica unatrag
<code>right(kut)</code>	<code>rt(kut)</code>	Zakreće pero za kut stupnjeva udesno
<code>left(kut)</code>	<code>lt(kut)</code>	Zakreće pero za kut stupnjeva ulijevo

Funkcije za relativna gibanja pera (vektorska grafika)

```
>>> from turtle import *
>>> title('Crtanje')
>>> pd()
>>> fd(200)
>>>
```



Modul `turtle` ima i funkcije za gibanje pera određeno apsolutnim koordinatama

Funkcija	Alternativni naziv	Opis djelovanja funkcije
<code>goto(x,y)</code>	<code>setpos(x,y)</code>	Pomiče pero na točku s koordinatama (x,y)
<code>goto(t)</code>	<code>setpos(t)</code>	Pomiče pero na točku s koordinatama t, gdje je t par brojeva (koordinate)
<code>setx(x)</code>		Postavlja prvu koordinatu na x, druga koordinata ostaje nepromijenjena
<code>sety(y)</code>		Postavlja prvu koordinatu na y, druga koordinata ostaje nepromijenjena
<code>setheading(kut)</code>	<code>seth(kut)</code>	Usmjerava pero tako da pokazuje u smjer kuta kut

Upravljanje perom i crtežom

Funkcije upravljanja perom

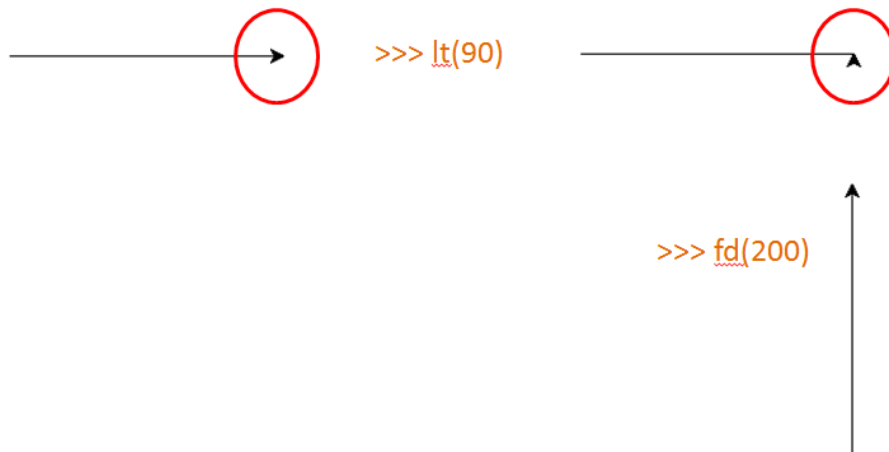
Funkcija	Alternativni naziv	Opis djelovanja funkcije
<code>pendown()</code>	<code>pd()</code> , <code>down()</code>	Pero se spušta i ostavlja trag
<code>penup()</code>	<code>pu()</code> , <code>up()</code>	Pero se podiže i ne ostavlja trag
<code>pensize(d)</code>	<code>width(d)</code>	Pero ima debljinu d jedinica
<code>showturtle()</code>	<code>st()</code>	Pero postaje vidljivo
<code>hideturtle()</code>	<code>ht()</code>	Pero postaje nevidljivo
<code>home()</code>		Postavlja pero u početni položaj
<code>undo()</code>		Poništava zadnju akciju pera; ta se funkcija može upotrijebiti višekratno

Funkcije upravljanja crtežom

Funkcija	Opis djelovanja funkcije
<code>reset()</code>	Briše sve crteže u grafičkom prozoru; postavlja pero u početni položaj s početnim atributima
<code>clear()</code>	Briše se crtež u grafičkom prozoru; pero ostaje nepromijenjeno

Gibanje pera

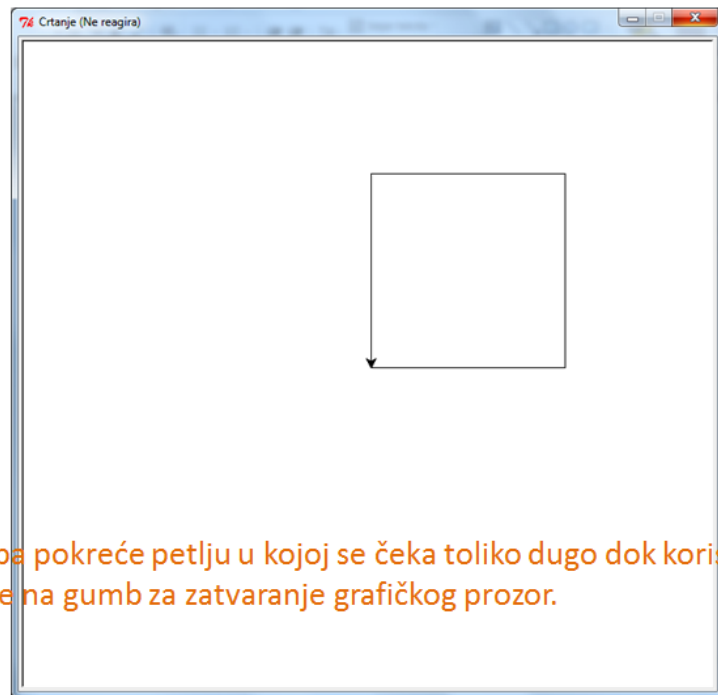
```
>>> from turtle import *
>>> title('Crtanje')
>>> pd()
>>> fd(200)
>>>
```



```

>>> from turtle import *
>>> title('Crtanje')
>>> pd()
>>> fd(200)
>>> lt(90)
>>> fd(200)
>>> lt(90)
>>> fd(200)
>>> lt(90)
>>> fd(200)
>>> mainloop() # ta naredba pokreće petlju u kojoj se čeka toliko dugo dok korisnik
                ne klikne na gumb za zatvaranje grafičkog prozor.

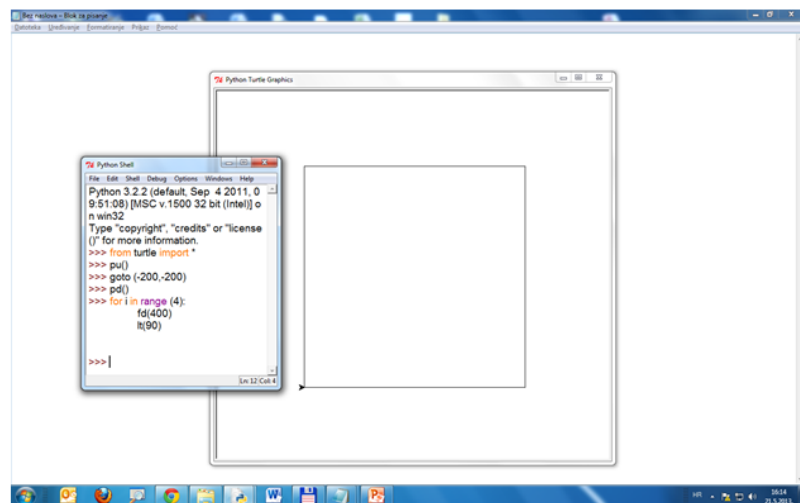
```



```

>>> from turtle import *
>>> pu()
>>> goto(-200,-200)
>>> pd()
>>> for i in range(4):
>>>     fd(400)
>>>     lt(90)
>>>

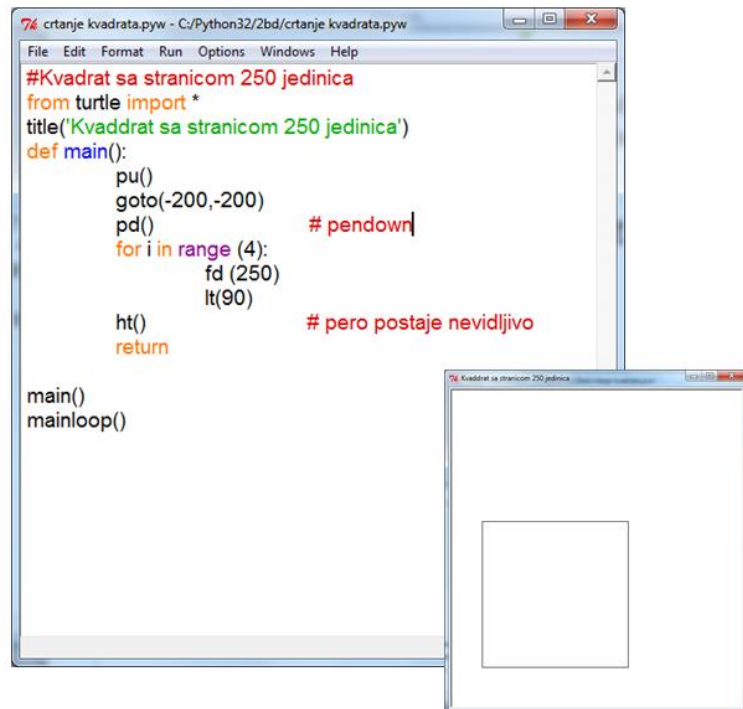
```



Napiši program koji će crtati kvadrat stranice duljine 250i dodati mu naslov: „Kvadrat sa stranicom 250 jedinica”, a na kraju će sakriti kornjaču.

```
#Kvadrat sa stranicom 250
jedinica
from turtle import *
title('Kvadrat sa stranicom
250 jedinica')
def main():
    pu()
    goto(-200,-200)
    pd() #
    pendown
    for i in range(4):
        fd(250)
        lt(90) #
    ht() #
    pero postaje nevidljivo
    return

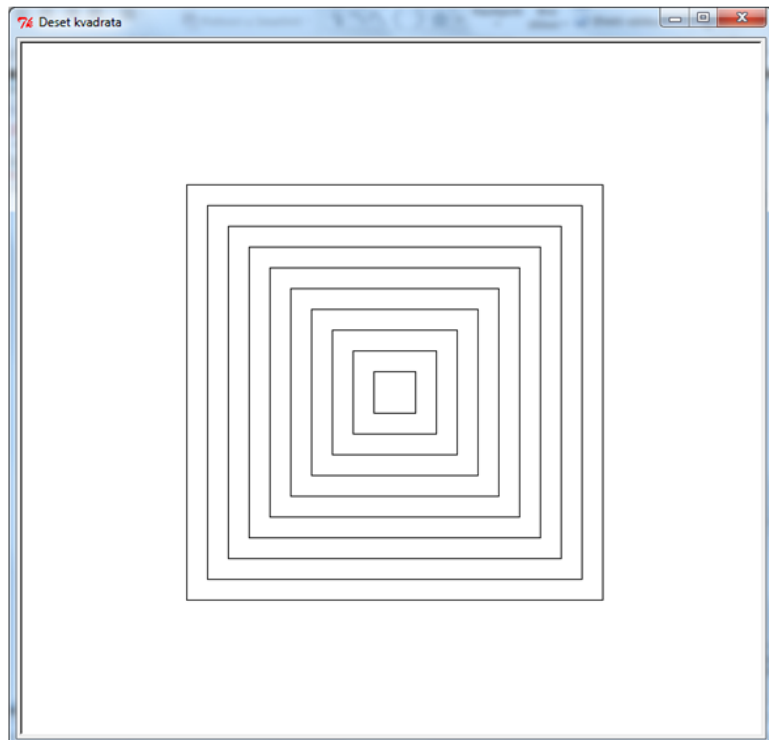
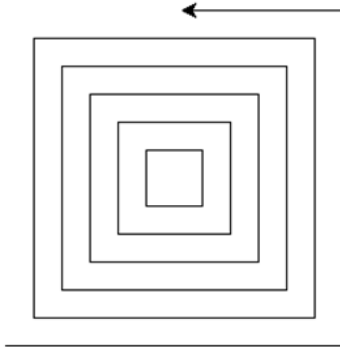
main()
mainloop()
```



```
#Deset kvadrata; primjer_08_02
from turtle import *
title('Deset kvadrata')
def kvadrat (stranica):
    pu()
    goto(-stranica // 2, -stranica // 2)
    pd()
    for i in range(4):
        fd(stranica)
        lt(90)
    return
def main():
    for i in range(10):
        stranica = (i + 1) * 40
        kvadrat (stranica)
    ht()
    return 'Gotovo!'

poruka = main()
print(poruka)
mainloop()
```





KOORDINATE GRAFIČKOG PROZORA

```
#Neke koordinate grafičkog prozora: primjer_08_07_A
from turtle import *

title('Koordinate nekih karakterističnih točaka')

def koord_točke(x, y):
    pu()
    goto(x, y)
    dot(5)
    sety(y - 20)
    ispis = '{0}, {1}'.format(str(x), str(y))
    write(ispis, align = 'center')
    return

def main():
    pu()
    x = [0, -200, 200, 200, -200]
    y = [0, -200, -200, 200, 200]
    for i in range(5):
        koord_točke(x[i], y[i])

    ht()
    return 'Gotovo!!'

poruka = main ()
print(poruka)
mainloop()
```

```
crtanje koordinata.pyw - C:/Python32/2bd/Grafika/crtanje koordinata.pyw
File Edit Format Run Options Windows Help
#Neke koordinate grafičkog prozora: primjer_08_07_A
from turtle import *

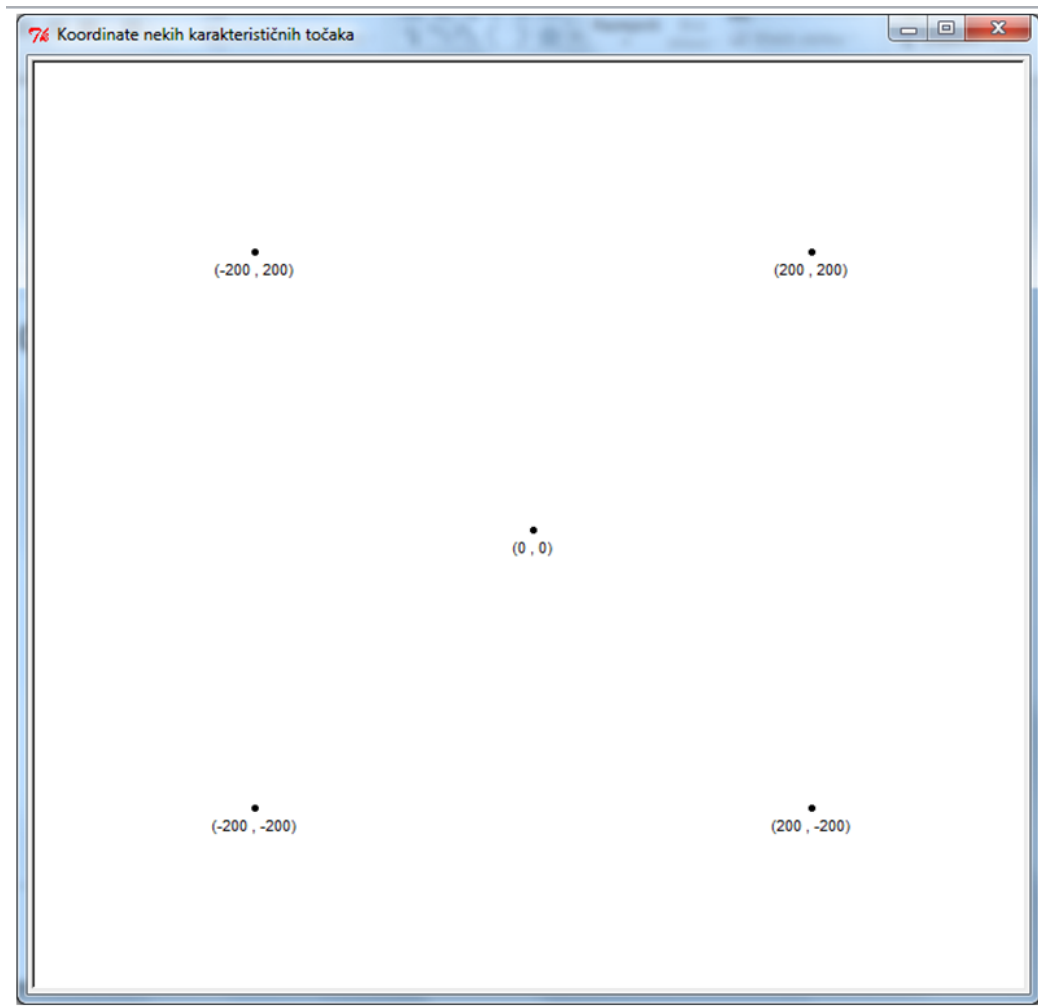
title('Koordinate nekih karakterističnih točaka')

def koord_točke(x, y):
    pu()
    goto(x, y)
    dot(5)
    sety(y - 20)
    ispis = '{0}, {1}'.format(str(x), str(y))
    write(ispis, align = 'center')
    return

def main():
    pu()
    x = [0, -200, 200, 200, -200]
    y = [0, -200, -200, 200, 200]
    for i in range(5):
        koord_točke(x[i], y[i])

    ht()
    return 'Gotovo!!'

poruka = main ()
print(poruka)
mainloop()
```

```
File Edit Format Run Options Windows Help
#Neke koordinate grafičkog prozora: primjer_08_07_A
from turtle import *

title('Koordinate nekih karakterističnih točaka')

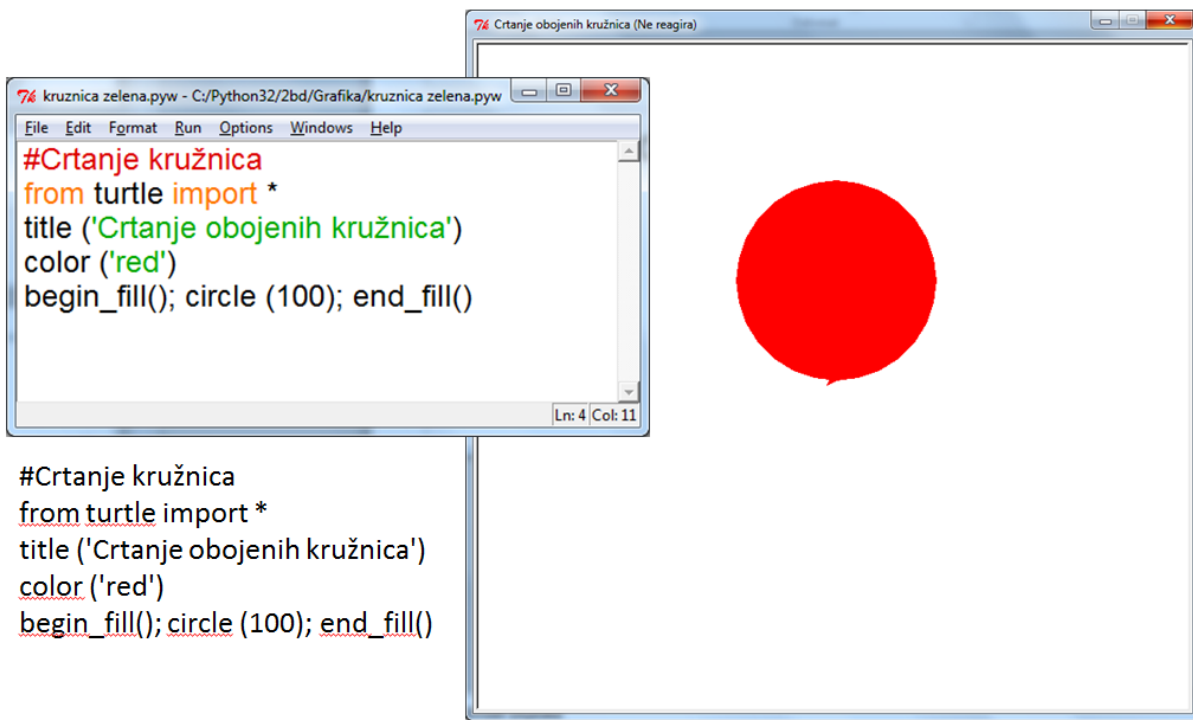
def koord_točke(x, y):
    pu()
    goto(x, y)
    dot(5)
    sety(y - 20)
    ispis = '{0}, {1}'.format(str(x), str(y))
    write(ispis, align = 'center')
    return

def main():
    pu()
    x = [0, -100, 200, 300, -150]
    y = [0, -100, -200, 300, 300]
    for i in range(5):
        koord_točke(x[i], y[i])

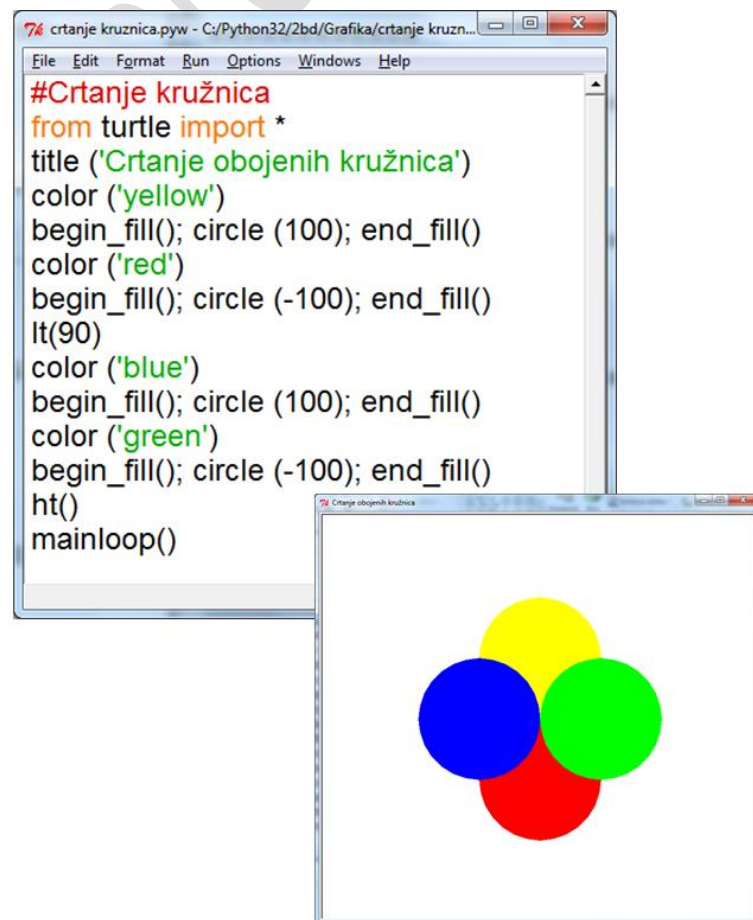
    ht()
    return 'Gotovo!!!'

poruka = main ()
print(poruka)
mainloop()
```

A screenshot of a Python window titled "Koordinate nekih karakterističnih točaka". The window displays a coordinate plane with five points plotted and labeled with their coordinates: (-150, 300), (300, 300), (0, 0), (-100, -100), and (200, -200). The points are arranged in a square pattern centered at the origin.



```
#Crtanje kružnica
from turtle import *
title ('Crtanje obojenih kružnica')
color ('yellow')
begin_fill(); circle (100); end_fill()
color ('red')
begin_fill(); circle (-100); end_fill()
lt(90)
color ('blue')
begin_fill(); circle (100); end_fill()
color ('green')
begin_fill(); circle (-100); end_fill()
ht()
# pero postaje nevidljivo
mainloop()
```



Bilješke s predavanja