Uvod u programiranje

Programiranje je vrlo zabavno i vjerujem da će vam se svidjeti.

Da bi smo mogli programirati moramo naučiti naredbe i načine njihovog korištenja.

Prvi korak u programiranju je razumijevanje samog problema kojeg želimo riješiti , a problem rješavamo pomoću dijagrama toka ili pseudojezika (pseudo kod).

Dijagram toka je grafički način rješavanja problema.

Pseudo jezik je tekstualni opis rješavanja problema i njega ćemo najviše koristiti pri rješavanju problema.

Kao što ste vidjeli na prošlim predavanjima:

Primjer_1:

Želimo napisati program koji će ispisati tvoje ime u jednom redu , a tvoje prezime u drugom redu

Pseudojezik	**Program**
Ispiši svoje ime Ispiši svoje prezime	print(" Ivica") print("Ivić")

Primjer_2:

Želimo napisati program za ispis tvoja dva najbolja predmeta u školi:

Pseudojezik	**Program**
Ispiši svoj prvi najbolji predmet Ispiši te svoj drugi najbolji predmet	print("Matematika") print("Hrvatski jezik")

Primjer _3:

Želimo napraviti program za zbrajanje dva broja i naravno ispisati rezultat .

Pseudojezik	**Program**
Unesite prvi broj a Unesite drugi broj b Operacija zbrajanja c=a+b Ispis rezultata zbrajanja c	a=int(input("Unesite prvi broj ")) b=int(input(" Unesite drugi broj")) c=a+b print("Rezultat zbrajanja je",c)

U zadnjem primjeru **(primjer_3),** brojeve smo zamijenili slovima, to ste radili u drugom razredu.

Kada brojeve zamjenjujete slovima u postupku programiranja ta slova nazivamo VARIJABLAMA.

Što je to varijabla ?

Uzmimo za primjer čašu za vodu. U toj čaši se može nalaziti voda, a kada popijete vodu u tu istu čašu možete staviti sok.

Znači u našoj zamišljenoj čaši u nekom trenutku možemo imati razne vrste napitaka, ali nikada više različitih napitaka u istom trenutku. (naravno ako poštujemo pravila).

U informatici i u postupku programiranja naša zamišljena čaša naziva se **varijabla**.

Znači varijabla je neka promjenjiva vrijednost koja se može po potrebi mijenjati i poprimati različite vrijednosti.

Pravila pisanja varijabli:

a) Varijable obavezno pišemo malim slovima engleske abecede
b) Varijabla može biti i jedna riječ
c) *Želite li koristiti dvije ili više riječi obavezno ih odvojite donjom crtom npr. petar_peric*
d) Varijabla ne smije početi sa brojem

Analizirajmo naš program, ali samo ona slova koja su označena crvenom bojom. **To su naše varijable.**

Program
a=3 b=5 c=a+b print("Rezultat zbrajanja je",c)

Varijabli a pridružili smo broj 3, a varijabli b broj 5.

Varijabla c nam služi da se u nju pohrani rezultat zbrajanja vrijednosti u varijabli a i vrijednosti u varijabli b.

Kada želimo ispisati rezultat zbrajanja naših varijabli, moramo prikazati vrijednost varijable c jer se u njoj nalazi rezultat zbrajanja.

Brojevi u varijablama ostaju pohranjeni sve dotle dok je program pokrenut ili ne u nesemo neku novu vrijednost.

Brojevi koji su pohranjeni u varijablama mogu nam poslužiti da sa tim brojevima napravimo razne matematičke operacije .

Pogledajmo primjer:

Program	**Pojašnjenje**
a=3 b=5 c=a+b d=a-b e=a*b print("Rezultat zbrajanja je",c) print("Rezultat oduzimanja je" ,d) print("Rezultat množenja je ", e)	U našem primjeru dodali smo još dvije varijable (d,e), naravno u varijabli d će biti pohranjen rezultat oduzimanja, a u varijabli e bit će pohranjen rezultat množenja. Da bi smo ispisali vrijednost neke varijable moramo koristiti naredbu print. To je ujedno i naša prva naredba za učenje.

Aritmetičke operacije u Python_u

Osnovne matematičke operacije	Koristit ćemo brojeve 8 i 3 – za primjer:
+ zbrajanje -oduzimanje * množenje / djeljenje // cjelobrojno dijeljenje % ostatak cjelobrojnog dijeljenja	8+3 = 11 8-3 = 5 8*3 = 24 8/3 =2.6666 8//3 =2 - cijeli broj 7%3 =1 - ostatak dijeljenja (2 i ostatak 1)

Vrlo važno: proučite sljedeće primjer:

Proučite sljedeće primjer			
16%4= 0 6%6= 0 6%1 =0 6%2 =0 6%3=0 6%4=2 6%5=1	5%6=5 4%6=4 5%6=5	16//4=4 16//5=3 16//4=4 16//3=5 16//2=8 16//1=16 16//16=1	4//16=0 5//16=0 12//16 =0

Pretvaranje matematičkih operacija u Python kod

a2+3b2 =ab	a*2+3*b*2=a*b	Važno : **Matematički zapis** **2b+3c** **U Pythonu obavezno mora biti u ovakvom zapisu:** **2*b+3*c**
$\frac{a+b}{2x}$	(a+b)/(2*x)	
8:4+2:2-3b	8/4+2/2-3*b	
6-6a+2b	6-6*a+2*b	
U matematici znamo redoslijed matematičkih operacija : **a) Zagrade b) Množenje ili dijeljenje c) Zbrajanje ili oduzimanje**		
Važno pravilo : *Kod operacija istih vrijednosti obavezno obratite pažnju da se operacije vrše s lijeva na desno:* **npr: 2*(6/2)/2*3 = Što je rezultat ? 9 ili 1 Odgovorite??**		

Zadaci za provjeru

Pitanje 1.

Dan je sljedeći Python izraz.
72 // 7

Što je od navedenog točno?

A. Oznakom // je predstavljena operacija cjelobrojnog dijeljenja brojeva 72 i 7 .
B. Oznakom // je predstavljena operacija dijeljenja brojeva 72 i 7.
C. Oznaka // ne predstavlja nikakvu operaciju u Pythonu.

Pitanje 2.

Dat je sljedeći Python izraz.
98 / 2

Koja je od gore navedenih tvrdnji istinita?

A. Oznakom / je predstavljena operacija cjelobrojnog dijeljenja brojeva 98 i 2.
B. Oznakom / je predstavljena operacija dijeljenja brojeva 98 i 2.

Pitanje 3.

Što je rezultat navedenog izraza?
4 // 2

Odgovor:

Pitanje 4.

Koji je rezultat navedenog izraza?
7 // 12

Odgovor:

Pitanje 5. *

Koji je rezultat sljedećeg izraza ?
12 / -6
A. -2
B. -2.0
C. Zapis nije pravilno napisan.

Pitanje 6.

Što će biti vrijednost sljedećeg izraza ?
32 % 24

Pitanje 7.

Neka je vrijednost varijable a prirodan broj djeljiv brojem 7. Što je od navedenog točno ?

A. Vrijednost a% 10 jednaka je 7.
B. Vrijednost a% 7 jednaka je 0.
C. Vrijednost a / 10 jednaka je 7.
D. Vrijednost // je jednaka 0.

Pitanje 8.

Što će biti rezultat sljedećeg izraza ?
15 % 10 + 15 / 3

A. 10
B. 10.0
C. 6.0
D. 6

Pitanje 9.

Što će biti rezultat sljedećeg izraza?
615 // 10

Pitanje 10

Što je rezultat sljedećeg izraza?
425 % 100

Pitanje 11.

Edita je dobila knjigu od 82 stranice. Odlučila je da krene sa čitanjem u ponedjeljak i da svakog radnog dana pročita jednak broj stranica. Ono što ne pročita do petka , pročitat će za vikend . Koliko je Edita pročitala stranica, a koliko će pročitati za vrijeme vikenda ?

ukupno = 82
m = ukupno // 5

Što predstavlja vrijednost dodijeljena varijabli m ?

A. Broj stranica koje će Editi ostati čitati tijekom vikenda.
B. Broj stranica koje će Edita pročitati tijekom jednog radnog dana.
C. Broj stranica koje će Edita pročitati za pet radnih dana.

Pitanje 12. *

Kako bi ste u Pythonu zapisali sljedeći matematički izraz ?
18 : (-15) + (-37) * 6

A. 18 / -15 + -37 * 6
B. 18 // -15 + -37 * 6
C. 18 / (-15) + (-37) * 6
D. 18: (-15) + (-37) * 6
E. 18: (-15) + (-37) * 6

Pitanje 13.
Koji od izraza je ispravno napisan u Pythonu ?

A. 35: (27 + 2)
B. (12 - 35) (27-2)
C. 35 * 27 * (12-27)
D. 12 - 35 x 27 - 2
E. 12 - [(35 + 27) - 2]

Pitanje 14. *

Koji je od sljedećih izraza ispravno napisan u Pythonu ?
A. 8475 - -9
B. 96 + 8 x 8
C. 475: 8-24
D. 8 475 + 24
E. 475 - -9 + 24

Pitanje 15.

Kako biste u Pythonu napisali sljedeći matematički izraz ?
$$\frac{2}{7} - 8$$
A. 2: 7 - 8
B. 2/7 - 8
C. 2: (7 - 8)
D. 2 / (7 - 8)

Pitanje 16. *

Kako bi ste u Pythonu zapisali sljedeći matematički izraz ?
$$-2 + \frac{4}{7} - 8$$
A. - (2 + 4) / (7 - 8)
B. - 2 + 4/7 - 8
C. - (2 + 4) / 7 - 8
D. - 2 + 4 / (7 - 8)

Pitanje 17. *

Što će biti rezultat sljedećeg matematičkog izraza ?
176 - 3 * (5 - 3)

Pitanje 18. *

Koji od navedena dva matematička izraza će dati veći rezultat ?

A. (8 + 3) * 6 + 15
B. 8 + 3 * 6 + 15

Pitanje 19.*

Napišite izraz koji predstavlja umnožak broja 42 i razliku brojeva 9 i 24.

A. 42 * 9-24
B. 42 * (9-24)
C. 42 - 9 * 24

Pitanje 20. *

Što će biti rezultat sljedećeg matematičkog izraza ?
16 * -3 + -2.0 * 4

A. -56,0
B. 56.0
C. -56
D. 56
E. Ispisat će se pogreška

Pitanje 21. *
Koji od sljedeća dva izraza će dati veći rezultat ?

A. 9 + 4 * (8 + 0)
B. 9 + 4 * 8 + 0
C. Vrijednosti danih izraza su jednake.

Pitanje 22.

Neka D predstavlja količinu novca koju ima Darko, a N iznos novca koji ima Nikola, što možete iščitati iz sljedećeg programskog koda ?

```
n = d + 2
```

A. Darko ima 2 Kune više od Nikole.
B. Nikola ima 2 Kune više od Darka.
C. Naredba nije pravilno napisana

Naredba PRINT

Pravilo pisanja naredbi , naredbe kao i varijable obavezno moraju biti pisane malim slovima.

Kao što smo vidjeli naša naredba print služi za ispis pohranjene vrijednosti u **varijablama**.

Naredba **print** ima i svoja pravila

1. Tekst koji želimo ispisati na ekranu obavezno mora biti pod navodnicima, ono što je pod navodnicima kako je napisano tako će se prikazati na ekranu, možemo koristiti bez ikakvih problema mala i velika slova kako god mi želimo.
2. Varijable ne smiju biti pod navodnicima ,
3. Naredbom **print** možemo urediti prikaz rezultata operacija nad varijablama.

Primjer_1.

Program	**Ispis na ekranu , vrijednosti varijabli su 3 i 2**
a=3 b=2 c=a+b print("Rezultat zbrajanja je",c) print("Rezultat zbrajanja broja",a,"i broja",b,"je",c) print(c)	Rezultat zbrajanja je 5 Rezultat zbrajanja broja 3 i broja 2 je 5 5

Koji je ispis ljepši ???? – prokomentirajmo !!

Naredba print može u sebi sadržavati razne matematičke operacije što će vam nekada dobro doći.

Program	**Objašnjenje**
a=3 b=2 print("Rezultat zbrajanja je",a+b)	U ovom primjeru nemamo varijablu za smještanje rezultata zbrajanja kao u gornjem primjeru. Operacija zbrajanja je u naredbi print i odmah će se ispisati , vrijednost zbrajanja se ne može koristiti u daljnjem programu.

Naredba **print** koristi još dva dodatna znaka /n za prelazak u novi red te /t tabulator za odvajanje riječi.

Primjer:

print("Pozdrav \n" , "Učimo Python\n")	Pozdrav Učimo Python
Print("Pozdrav\t","Učimo Python\t")	Pozdrav Učimo Python

Zadaci za vježbu :

Kada varijabli pridružujete tekst , on obavezno mora biti pod navodnicima

1. Napiši program koji varijabli x pridružuje tvoje ime(npr. Luka) i ispisuje poruku: Zovem se x.

Rješenje:

Pseudojezik	Programski kod
Varijabli x pridruži svoje ime Ispiši tekst (zovem se x)	x="LUKA" print(" Zovem se ",x)

Zadatak_1:

Varijabli a pridružite svoje ime, a varijabli b svoje prezime. Ispišite tekst moje ime je a ,a moje prezime je b. moje ime je "a", a moje prezime je "b"

Zadatak_2:

Varijabli x pridruži svoje ime(npr. Luka), a varijabli g broj svojih godina(npr. 11) te ispiši poruku: Zovem se x i imam g godina.

Zadatak_3:

Varijabli a pridruži broj 2, a varijabli b broj 7. Ispiši - umnožak broja a i broja b je "rezultat".

Zadatak_4:

Varijablama a,b i c pridruži tri broja po želji i izračunaj i ispiši umnožak ta tri broja (npr. 5,8,3).

Zadatak_5:

Varijabli n pridruži jedan paran broj(npr.2), te ispiši sljedeći po redu paran broj.

Zadatak_6:

Varijabli n pridruži broj 8 te ispiši trostruko veći broj od njega.

Naredba INPUT

Naredba INPUT nam služi za upis vrijednosti u varijablu, ***prisjetite se da smo rekli da naredba print služi za ispis vrijednosti varijable.***

Pravila za naredbu input:

Pošto naredbu input koristimo za unos vrijednosti u varijablu tj. preko tipkovnice možemo unijeti slova ili brojeve.

*U naredbi **input** moramo naznačiti što unosimo.*

Naredba input obavezno počinje s nekom varijablom kako bi u nju mogli pohraniti neku vrijednost koju unosimo preko tipkovnice.

Kod unosa brojeva moramo naznačiti da li se radi o cijelom broju ili decimalnom broju.

Za cijeli broj koristimo naredbu : int
Za decimalne brojeve koristimo naredbu : float
Za unos teksta naredba input *nema nikakvih dodataka*

Program	
a=int(input("Unesite prvi broj ")) b=float(input(" Unesite drugi broj")) x=input("unesi svoje ime") c=a+b print("Rezultat zbrajanja je",c)	1. Cijeli broj npr. 2,4 6,10,101. 2. Decimalni broj 2.3 ,3.555 ,5.54 3. Običan tekst **Kod decimalnih brojeva ne koristimo zarez, koristimo točku.** ***Važno – koliko ste zagrada otvorili toliko zagrada morate i zatvoriti.***

Primjeri zadataka :

Zadatak_1:
Napiši program koji unosi (naredba input) tvoje ime i ispisuje pozdravnu poruku: Pozdrav, x!

```
x=input(" Unesite svoje ime")
print("Pozdrav !",x)
```

Zadatak_2:
Napiši program koji unosi (naredba input) tri cijela broja a,b,c i ispisuje njihov umnožak

```
a=int(input(" unesi prvi broj   "))
b=int(input("unesi drugi broj   "))
c=int(input(" Unesi treći broj  "))
d=a*b*c
print(" Rezultat umnoška tri broja je ",d)
```

Želimo li unositi decimalne brojeve, umjesto int koristit ćemo float.

Napišimo program za zbrajanje dva decimalna broja. Brojeve unosimo preko tipkovnice.	**A=float(input("Unesi prvi broj := ")) B=float(input("Unesi drugi broj := ")) c=a+b print("Zbroj brojeva ",a,"i",b,"je",a+b))**

Prikaz decimalnih brojeva

Python decimalne brojeve prikazuje sa do 16 decimalnih mjesta.
To može biti dosta nepregledno pri ispisu takvih decimalnih rezultata.
Ispis decimalnih brojeva možemo urediti tako da nam se ispisuje rezultat sa onoliko decimalnih mjesta koliko mi želimo.

U tu svrhu koristimo formatirani ispis s kojim proširujemo naredbu print naredbom .format

Primjer_1:

Prikaz varijable x zaokruži na tri decimale
x=2.3456789

```
print("Zaokruži na tri decimale{:5.3f}".format(x))
```

a=4 b=5 print("Zbroj brojeva",a,"i",b,"je",a+b)	Ovaj ispis smo već obradili u naredbi print
a=4 b=5 c=a/b print("Zaokruži na dvije decimale{:5.2f}".format(c))	{:5.2f}".format(c) – 2f – označava 2 decimalna mjesta {:5} Znači da smo za prikaz decimalnog mjesta osigurali pet(5) mjesta u to ubrajamo i decimalnu točku. Uzmimo broj 10/3 =3.33333333 03.33 - {:5.2f} rezervirali smo pet mjesta od kojih su dva decimalna , nula s lijeve strane broja nema nikakvo značenje. To smo učili na matematici.

Zadatak_3:
Napiši program koji unosi (naredba input) dva broja te ispisuje njihov zbroj, razliku, umnožak i količnik, jedno ispod drugoga, Količnik zaokružite da dvije decimale.

Zadatak_4:
Napiši program koji unosi (naredba input) jedan parni broj x i ispisuje sljedeći parni broj.

Zadatak_5:
Ana i Marija bile su skupa na moru i na plaži su skupljale školjke. Ana je skupila n školjaka, a Marija tri puta više od Ane. Napiši program koji će na temelju ulaznih podataka izračunati i ispisati koliko školjaka je skupila Marija.

Npr: Koliko školjaka je skupila Ana?12
Marija je skupila 36 školjaka.

Zadatak_6:
Fran i Jan idu na sladoled. Četiri kuglice sladoleda Fran je platio n kuna. Napiši program koji će na temelju ulaznih podataka izračunati i ispisati koliko kuna košta jedna kuglica sladoleda.

UVJETNE NAREDBA IF-ELSE—ELIF

Što moramo znati??: naredba za ispis je print(), “=” je znak pridruživanja vrijednosti varijabli. Postoji više načina ispisa (poželjno da znate barem jedan), a aritmetičke operatore ste obavezni znati.

Dosad smo u programima koristili aritmetičke operatore. Međutim, što učiniti ako trebamo u programu uspoređivati više vrijednosti ?

Tu nam pomažu operatori uspoređivanja.

Ima ih nekoliko, također ih već znate iz matematike, ali u matematici se oni pišu malo drugačije u odnosu na Python. U nastavku se nalazi prikaz operatora.

Značenje	Oznaka u matematici	Oznaka u Pythonu
Manje od	<	<
Veće od	>	>
Manje ili jednako	≤	<=
Veće ili jednako	≥	>=
Jednako	=	==
Različito	≠	!=

NAPOMENA: Pazite na razliku između “=” i “==”. **Znak** “=” je znak pridruživanja koji služi da se neka vrijednost pridruži varijabli, dok znak “==” predstavlja operator uspoređivanja koji je istinit ako su uspoređene vrijednosti jednake.

Možemo dalje ?. Naime, u programu mogu postojati uvjeti u kojima se program grana na više mogućih rješenja ovisno o uvjetu.

Primjer_1:

Evo primjera iz stvarnog života; dođemo u dućan sportske opreme i kupujemo sve što nam treba za dječje igralište . Na kraju kupnje dođemo na blagajnu i u slučaju iznosa većeg od 5000 kn imamo pravo na besplatnu dostavu što se ispisuje na računu. Upravo ćemo ovaj zadatak riješiti, ali ćemo ga malo pojednostaviti. Kako ne bismo unosili svaku stavku posebno, unijet ćemo samo konačnu vrijednost kupovine, a program nam treba ispisati: vrijednost računa, pravo na dostavu (samo u slučaju vrijednosti računa većeg od 5000 kn) te prigodnu rečenicu (“Dođite nam opet!”).

Sintaksa za if

If uvjet:
blok naredbi

#Unos vrijednosti računa
racun=float(input("Unesite vrijednost računa"))
#Uvjet if i ispis prigodne rečenice
if racun>5000:
    print("Imate pravo na besplatnu dostavu")
print("Dođite nam opet")
	# - koristimo za komentare ,oni služe za lakše snalaženje u programu i ne utječu na izvršavanje programa.
	Ono što je u uvjetu if ,mora biti uvučeno, najbolje koristite tipku tab na tipkovnici.
	Ovaj dio naredbe nije vezan za uvjet if

Kao što vidite na primjeru, if uvjet sastoji se od ključne riječi **if**, uvjeta i nakon toga slijedi znak ":". Važno je nakon toga naglasiti da sve što spada pod naredbu IF se nalazi uvučeno i to će se izvršiti **samo ako je uvjet zadovoljen**.

Kao uvlaka se preporučuje tipka tab i to uređen kao četiri razmaka.

Naravno, to je najjednostavnije korištenje if uvjeta. Sada zamislimo sljedeći slučaj. Došli smo u školu i trebamo napraviti program koji će nam ispisivati da li smo prošli ili pali ispit. To ćemo učiniti tako da napravimo program koji će u slučaju ocjene 1 ispisivati "Niste položili ispit.", a za ostale slučajeve ispisivati "Položi ste ispit ! Čestitamo"

Sintaksa za if – else:

if uvjet:
    blok naredbi
else:
    blok naredbi

# unos ocjene 	
ocjena=int(input("Unesite ocjenu "))
#Uvjet za if – else
if ocjena==1:
    print("Niste položili ispit.")
else:
    print("Položili ste ispit ! Čestitamo")
	Kao što vidite u uvjet if koristimo znak jednakosti ==.
	Ako je uvjet točan izvršit će se onaj dio programa koji je u uvjetu if.
	Ako uvjet nije točan izvršit će se onaj dio programa koji se nalazi ispod else. *Važno* je zamijetiti da se naredbe ispod if i else moraju uvući.

Programski primjeri s rješenjima:

Napiši program koji unosi dva broja te ispisuje poruku jesu li brojevi jednaki ili različiti.	*#program za unošenje i usporedbu dva unesena broja*
	a=int(input("Upiši prvi broj:"))
	b=int(input("Upiši drugi broj:"))
	if a==b:
	    print("Brojevi su jednaki")
	else:
	    print("Brojevi su različiti")

Primjer_2:

Napiši program za unos dva broja (input) te ispisuje poruku koji od ta dva broja je veći.	**#usporedba unesenih brojeva**
	a=int(input("Unesi prvi broj:")
	b=int(input("Unesi drugi broj:")
	if a>b:
	    print("broj",a,"je veći")
	else:
	    print("broj",b,"je veći")

Primjer_3:

Napišite program koji unosi duljine stranica a i b te ispisuje poruku da li se o kvadratu ili pravokutniku.
Važno:
Što je karatteristično za kvadrat ??
	a=int(input("duljina stranice a:"))
	b=int(input("duljina stranice b:"))
	if a==b:
	    print("KVADRAT")
	else:
	    print("PRAVOKUTNIK")

Kao što možete vidjeti, za uspješno rješavanje zadataka sa uvjetom IF – ELSE bitno je poznavati matematičke operatore uspoređivanja.(>,<,==,!=..) i aritmetičke operatore (+,-,*,/, //, %....)..

Zadaci za vježbu :

Zadatak_1:

Napiši program koji unosi jedan broj(input)te ispisuje je li taj broj veći ili manji od 100.

Zadatak_2:

Napiši program koji unosi jedan broj (input) te ispisuje poruku je li učitani broj paran ili ne.

Kada je neki broj paran ??, koji ćete aritmetički operator koristiti ??

Zadatak_3:

Napiši program koji unosi jedan broj te ispisuje poruku je li učitani broj djeljiv s 5 ili ne.

Sintaksa za if – elif – else: *višestruki uvjeti:*

If uvjet:

blok naredbi

elif uvjet:

blok naredbi

else:

blok naredbi

Zamislimo situaciju da ovisno o unesenoj ocjeni od 1 do 5, program ispisuje poruku ocjenu koju ste dobili i da li ste ili niste prošli zamišljeni ispit.

# unos ocjene 	
ocjena=int(input("Unesite ocjenu: "))
# uvjet if – elif – else 	
If ocjena==1:
    print("Ocjena ispita je nedovoljan, niste prošli")
elif ocjena==2:
    print(" Ocjena ispita je dovoljan, prošli ste")
elif ocjena==3:
    print("Ocjena ispita je dobar,prošli ste")
elif ocjena==4:
    print("Ocjena ispita je vrlo dobar,prošli ste")
elif ocjena==5:
    print("Ocjena ispita je odličan, prošli ste")
else:
    print(" Unijeli ste krivu ocjenu")
	elif – kao što možete vidjeti na ovom primjeru koristimo višestruke uvjete za prikaz svake ocjene od 1 do 5.
	
	Ako ocjena nije u rasponu od 1 do 5 aktivirat će se događaj pod naredbom else.

Primjeri programa s rješenjima:

Primjer_1:

Napiši program koji unosi rezultat nogometne utakmice za Brazila i Hrvatsku te ispisuje tekst "Brazil je pobjednik" u slučaju da je Brazil pobijedila, tekst "Hrvatska je pobjednik" u slučaju da je Hrvatska pobijedila ili tekst "Neriješeno" u slučaju da su obje ekipe osvojile jednak broj bodova.	h=int(input(" Broj golova hrvatske: "))
	s=int(input("Broj golova brazila: "))
	if h>s:
	    print("Hrvatska je pobjednik")
	elif h<s:
	    print("Brazil je pobjednik")
	elif h==s:
	    print("Neriješeno")

Kao što vidite u ovom primjeru uvjeti mogu završiti i bez naredbe else:

Primjer_2:

Napiši program koji prevodi boje (plava, zelena, crvena) s hrvatskog na engleski jezik. Ukoliko upišemo nešto drugo osim te tri boje, pojavit će se poruka: Ta boja ne postoji u programu.	**b=input("Upisi boju:")
	if b=="plava":
	    print ("plava – blue")
	elif b=="zelena":
	    print ("zelena – green")
	elif b=="crvena":
	    print ("crvena – red")
	else:
	    print ("Ta boja ne postoji u programu!")**

Logički operatori:

Sada smo prošli uvjet if i njene varijacije. Međutim, u programiranju se znaju javljati zadaci koji zahtijevaju da neki uvjeti budu povezani tako da se zadatak izvrši samo ako su oba uvjeta zadovoljena i sl. Tu na scenu stupaju logički operatori.

Logički operatori su operatori koji ovisno o uvjetu vraćaju **True** (istinu) odnosno logički 1 i **False** (laž) odnosno logičku 0. U Pythonu postoje logički operatori **and** (i), **or** (ili) i **not** (ne).

Logički operator **and** je operator koji vraća vrijednost **True** (istinu) ako i samo ako su svi uvjeti istiniti (zadovoljeni). U suprotnome će vraćena vrijednost biti **False** (laž).

Tablica istinitosti za AND			**Tablica istinitosti za OR**		
Uvjet 1.	**Uvjet 2.**	**rezultat**	**Uvjet 1.**	**Uvjet 2.**	**rezultat**
1	**0**	**0**	**1**	**0**	**1**
0	**1**	**0**	**0**	**1**	**1**
0	**0**	**0**	**1**	**1**	**1**
1	**1**	**1**	**0**	**0**	**0**
1 je TRUE	**0 je FALSE**				

VAŽNO:

Kod logičkog operatora AND svi uvjet moraju biti zadovoljeni da bi uvjet bio točan

Kod logičkog operatora OR dovoljno je da jedan od uvjeta bude točan.

Primjer zadataka s logičkim operatorom

Napiši program koji unosi dob putnika te ispisuje poruku o cijeni karte prema pravilima: * osobe mlađe od 8 godina voze se besplatno * osobe od 8-18 godina plaćaju kartu 5 kn * osobe starije od 65 godina plaćaju kartu 1 kn * sve ostale osobe plaćaju kartu 8 kn	**a=int(input("Upiši starosnu dob putnika: "))
	if a<8:
	    print("karta je besplatna")
	elif a>=8 and a<18:
	    print("karta je 5kn")
	elif a>65:
	    print("karta je 1kn")
	else:
	    print("karta je 8kn")**

Primjer_2:

Unesite vrijednosti za tri stranice trokuta i usporedite da li je trokut jednakostraničan ili raznostraničan ili jednakokračan	**a=int(input("Vrijednost stranice a: "))
	b=int(input("Vrijednost stranice b: "))
	c=int(input("Vrijednost stranice c: "))
	If a==b and a==c:
	        print("Jednakostraničan")
	elif a==b or a==c or b==c:
	        print("Jednakooračan")
	else:
	        print("Raznostraničan")**

Primjer_3:

Napišimo program koji će nam omogućiti unos godine i ispisati da li je unesena godina prijestupna ili ne.	**god=int(input("Unesi godinu "))
	If (g%4==0 and g%100 !=0) or (g % 400==0):
	        print("Prijestupna")
	else:
	        Print("Nije prijestupna")**

Primjeri zadataka za provjeru znanja:

Pitanje 1.

Što će biti rezultat izvršavanja sljedećeg programskog koda?

```
semafor = 'plavo'
if (semafor == 'zeleno'):
       print('pređi ulicu')
if (semafor == 'crveno'):
      print('ne možeš preči ulicu')
```

a) Ispisat će se tekst "ne možeš priječi ulicu" .
b) Ispisat će se tekst "prođi ulicu".
c) Ispisat će se tekst "plavo".
d) Nijedan od ponuđenih odgovora nije točan.

Pitanje 2.*

Što će biti rezultat izvršavanja sljedećeg programskog koda?

```
a = -1
b = -1
if (a > b):
  print(a)
else:
  print(b)
```

Odgovor:

Pitanje 3. *

Napišimo program koji će tražiti da unesemo cijene dvije vrste sladolede (“čokolada“,“jagoda“) ovisno o unesenoj cijeni ispisat ćemo koji je od njih skuplji, ako je skuplja čokolada ispisat će se “čokolada“ , ako je skuplja jagoda ispisat će se “Jagoda“.

```
cokolada=int(input('Unesi cijenu za čokoladu))
jagoda=int(input('Unesi cijenu za jagodu))
if (cokolada>jagoda):
  print("čokolada")
else:
  print("jagoda")
```

a) čokolada
b) jagoda
c) Ispisat će se imena oba sladoleda.
d) Program neće ispisivati nikakvu poruku.

Pitanje 4. *

Što bi ste od ponuđenog uzeli i stavili u uvjet IF da bi ste dobili ispravan rezultat

```
a = int(input('Unesite jedan broj'))
if (   ):
      print('Broj je djeljiv sa 5')
else:
      print('Broj nije djeljiv sa 5')
```

a) a% 5 == 0
b) a / 5 == 0
c) a. 5 == 0
d) Niti jedan od gore navedenih odgovora nije točan.

Pitanje 5.

Što će se ispisati nakon izvršenja sljedećeg koda?

```
a = 8
print( (a < 6) or (a > -10) )
```

a) Istina
b) Netočno

Pitanje 6.

Što će biti rezultat izvršavanja sljedećeg koda?

```
a = 2
b = 62
if (( a >= 10) or (b <= 70)) and (a + b > 50):
     print(a - b)
else:
     print(2 * a - b)
```

Odgovor:

Pitanje 7.

Što će biti rezultat programa nakon izvršenja sljedećeg koda?

```
a = 17
print( (a < 6) and (a > -10) )
```

a) Istina
b) Netočno

Pitanje 8.

Što će se ispisati nakon izvršenja sljedećeg koda?

```
a = 8
print( (a < 6) or (a > -10) )
```

a) Istina
b) Netočno

Pitanje 9.

Što će biti rezultat izvršavanja sljedećeg koda?

```
a = 2
b = 62
if (( a >= 10) or (b <= 70)) and (a + b > 50):
  print(a - b)
else:
  print(2 * a - b)
```

Odgovor:

Pitanje 10.

Cijena autobusne karte iznosi 65 Kn. Za djecu(stariju od 7 godina i ne stariju od 20 godina) i penzionere (ne mlađe od 65 godina), odobrava se popust od 10 Kn. Napišite program gdje će ovisno u unesenoj godini života ispisati i cijena karte.

Kojim uvjetom bi ste nadopunili programski kod

```
godine = int(input("Unesi koliko imaš godina"))
cijena = 660
if (________________________):
  cijena = 660 - 100
print(cijena)
```

a) (godina> 7 and godina <= 20) or (godina> 65)
b) (godina> 7 and godina <20) or (godina> 65)
c) godina> 7 and godina <= 20 or godina> = 65
d) godina> 7 and godina <20 or godina> 65

Pitanje 11. *

Što će se ispisati prilikom izvršavanja sljedećeg koda ?

```
bodovi = int(input('Unesi broj bodova postignutih na ispitu'))
if bodovi > 85:
  o = 5
elif bodovi > 70:
  o = 4
elif bodovi>55:
  o = 3
elif bodovi>39:
  o = 2
else:
  o = 1
print(o)
```

ako se kao broj bodova unese broj 89 .

Pitanje 12.

Što će se ispisati prilikom izvršavanja sljedećeg koda?

```
a = 2
b = 10
if (a + b > 10):
    print(a * a)
elif (a + b == 10):
    print(a-b)
else:
    print(b)
```

Odgovor:

Pitanje 13.*

Što će se ispisati prilikom izvršavanja sljedećeg koda?

```
a = -10
b = -8
c = -1
if (c > 10):
    print(a * a)
elif (a + b > 10) or (b % 2 == 0):
    print(a - b)
else:
    print(b)
```

Odgovor:

UVOD U Petlje

Prošli smo put radili uvjete, odnosno grananje programa u ovisnosti o nekom uvjetu ili stanju u programu. Danas ćemo obraditi (petlje, ponavljanje dijela programa), koje su vrlo jednostavne.

Što je uopće petlja i čemu služi?

Petlja predstavlja dio programa koji možemo ponoviti određen broj puta. Na taj si način olakšavamo posao i nemamo potrebe pisati neki dio programa više puta.

Najjednostavniji primjer petlje je onaj školski; napišite 5 rečenica „Volim programiranje!“. Iako je to relativno mali broj ponavljanja, nekima neće biti problem i napisat će pet redova i u svakom ispisati rečenicu s pomoću funkcije print().

Međutim što ako umjesto broja 5 piše broj 50?

Naravno, možete se koristiti copy/paste metodom i brojiti redove, ali čemu se mučiti i filozofirati? Upravo zato upotrebljavamo petlje.

Postoji više vrsta petlji, a mi ćemo naučiti dvije (for i while) petlje, odnosno petlju s brojačem (*for*) i petlju s uvjetom (*while*).

Prije nego što krenemo rješavati zadatke, hajde da na ovom prethodnom primjeru prikažemo upotrebu petlji. Prvo da vidimo kako bi to izgledalo s pomoću *for* petlje:

```
for brojac in range(1,6):
    print(brojac)
```

Na početku vidimo ključnu riječ *for*, koja nam objašnjava da se radi o petlji s brojačem.

Nakon toga vidimo **VARIJABLU** brojac koja će poprimati vrijednosti iz funkcije range(). Unutar funkcije range() vidimo vrijednosti 1 i 6 odvojene zarezom. Ovdje treba napomenuti da je prva vrijednost početna, a druga završna, s tim da se ponavljanje izvršava do završne, **a nju ne uključuje**.

U ovom bi primjeru varijabla brojac u prvom koraku imala vrijednost 1 i zatim bi se s pomoću funkcije print() ispisao broj 1. Nakon toga se opet vraćamo na početak petlje i varijabla ima vrijednost 2 te se opet to ispisuje. I tako sve do vrijednosti 6, gdje brojač dobiva upravo tu vrijednost. Budući da prema funkciji range() taj broj ne ulazi u interval, ne ulazi se u petlju, nego se izlazi van. I na taj način smo ispisali brojeve od 1 do 5.

Kod petlje s uvjetom, odnosno petlje while, obavezni smo sami napraviti brojač jer se u zaglavlju petlje zapravo ispisuje uvjet, to jest, u ovom slučaju, je li naš brojač manji od 6, a to izgleda ovako:

```
brojac = 1
while brojac < 6:
    print(brojac)
    brojac = brojac + 1
```

Za početak, prije same petlje, moramo napraviti varijablu brojac u kojoj se nalazi naša početna vrijednost, u ovom slučaju to je broj 1. Nakon toga slijedi ulazak u petlju. Vidimo ključnu riječ while, koja označuje početak while petlje, a nakon toga slijedi uvjet gdje se u slučaju ako je uvjet zadovoljen ulazi u petlju, a u suprotnome izlazi iz nje.

Pri prvom koraku petlje vrijednost brojača je 1 i to je manje od 6, pa se ulazi u petlju i ispisuje vrijednost brojača. Nakon toga se brojaču povećava vrijednost za 1.

Ovo je vrlo važno jer bi u protivnom vrijednost brojača ostala 1, tj. bila bi konstanta i u tom slučaju bi **uvjet** uvijek bio zadovoljen i petlja bi se stalno izvodila što nazivamo beskonačnom petljom. Ipak, ovdje smo povećali vrijednost varijable brojača koja se izvršava sve dok je brojač manji od 6. Kada vrijednost brojača dođe do broja 6, dolazimo do zaglavlja petlje, ali uvjet nije ispunjen pa se zbog toga izlazi iz petlje.

PETLJA FOR

Kao što smo do sada naučili to je konačna petlja koja ima svoj početak i kraj.

Petlja for ima više oblika i prikazat ćemo sve mogućnosti.

U petlji for brojevi i rezultati obrade uvijek se ispisuju u stupcu tj. jedan ispod drugoga

Oblik_1: Imamo definiranu početnu i završnu vrijednost

for a in range(1,6): print(a)	a – predstavlja brojač koji nam poprima vrijednost za jedan više svaki puta kada prolazi kroz petlju do konačno broja 6. koji je naveden u funkciji range(1,6). Naredbe koje nalaze u petlji obavezno moraju biti uvučene, to smo učili kod uvjeta.

Oblik_2: Definiramo početnu vrijednost, završnu vrijednost i korak uvećanja

for a in range(1,6,2): print(a)	(1,6,2) – Prvim prolaskom kroz petlju ispisat će se broj 1, drugim broj 3, sljedećim prolaskom broj 5. Ispisat ćemo <u>neparne brojeve</u> od 1 do 6. Što ako želimo ispisati <u>parne brojeve</u> ? Stavit ćemo samo range(0,6,2)

Oblik_3: Ispis brojeva od većeg prema manjem

for a in range(6,1,-1): print(a)	(6,1,-1) – plavi -1 – predstavlja da idemo u natrag od većeg prema manjem(5,4,3,2,1) Razmislite !! Što bi ste promijenili da ispišete neparne brojeve od 6 do 1, a što da ispišete parne brojeve od 6 do 1)??

Oblik_4: Ispis brojeva u retku

for a in range(1,6): print(a,end=" ")	end="" – omogućuje ispis rezultata obrade u retku. Važno: ako stavite razmak između navodnika, dobit ćete i lijepi razmak između brojeva u retku . Stavite li između navodnika (- ili ,) brojevi će biti međusobno odijeljeni tim znakom.

Oblik_5: Petlja for i zamjena naredbe range sa in i not in

a="sunce" for i in a: print(i)	Umjesto range , koristimo ključnu riječ in. In – provjerava da li se ono što tražimo nalazi u stringu.(znakovnoj varijabli).

Primjeri s rješenjima petlja for i petlja while

Primjer_1:

Napiši program koji 5 puta ispisuje tvoje ime pomoću petlje for	for i in range(5): print (Petar")

	a=0
	while a<5:
	    print("Petar")
	    a=a+1

Primjer_2:

Napisi program koji 4 puta ispisuje tvoje ime pomoću petlje for i zatim jednom tvoje prezime ispod svih imena.	for i in range(4): print ("Petar") print ("Petrović") Pojašnjenje: Ime želimo ispisati četiri puta i ono mora biti u petlji , zato je uvučeno. Prezime je van petlje i zato nije uvučeno.

	a=0
	while a<5:
	    print("Petar")
	    a=a+1
	print("Petrović")

Primjer_3:

Napiši program koji pomoću petlje for ispisuje sve parne brojeve od 40 do 60 jedan pored drugoga.
 	
Napišite program za ispis neparnih brojeva od 20 do 40 , jedan ispod drugog.
	for i in range(40,61):
	    if i%2==0:
	        print(i, end = " "´)
	
	Važno:
	Obratite pažnju na uvlake , if je u petlji i zato je uvučen, a ispis je u okviru uvjeta if i uvučen je u okviru nje. tj. dvostruko je uvučen u odnosu na petlju for.

	a=40
	while (a>39) and (a<=60):
	    if a%2==0:
	        print(a,end=" "))
	    a=a+1

Primjer_4:

<table>
<tr><td rowspan="2">Napiši program koji pomoću petlje for ispisuje brojeve djeljive s pet od 100-150 jedan pored drugoga.</td><td>for i in range(100,151):
    if i%5==0:
        print(i, end = "")</td></tr>
<tr><td>a=100
while (a>99) and (a<=150):
    if a%5==0:
        print(a,end=" "))
    a=a+1</td></tr>
</table>

Primjer_5:

<table>
<tr><td rowspan="2">Napiši program za ispis brojeva od 1 do 10, ali bez broja 6.</td><td>for i in range(1,11):
    if i!=6:
        print(i, end = "")</td></tr>
<tr><td>a=0
while a<11:
    if a!=6:
        print(a,end=" ")
    a=a+1</td></tr>
</table>

Primjer_6:

<table>
<tr><td>Napiši program koji ispisuje slovo po slovo, jedno ispod drugoga, pomoću naredbe for riječ ‘sunce’</td><td>a="sunce"
for i in a:
    print(i)</td></tr>
</table>

Primjer_6a:

<table>
<tr><td>Napiši program koji za upisanu riječ ispisuje samo samoglasnike.</td><td>a=input("Upiši riječ:)
for i in a:
    if i in "aeiouAEIOU":
        print (i)</td></tr>
</table>

Primjer_7:

<table>
<tr><td>Proširi prethodni program tako da umjesto samoglasnika program prebroji i ispiše broj samoglasnika u rijeci.</td><td>a=input("Upiši riječ:")
sam=0
for i in a:
    if i in "aeiouAEIOU":
        sam=sam+1
print ("Broj samoglasnika:", sam)</td></tr>
<tr><td colspan="2">Važno: U ovom primjeru za korištenje brojača moramo definirati početnu vrijednost sam=0 i uvećanje te vrijednosti za 1 sam=sam+1</td></tr>
</table>

Primjer_8:

Zbrajanje u petlji FOR
Napišite program koji ispisuje zbroj parnih brojeva od 1 do 20.
	zbroj = 0
	for i in range (0, 21, 2):
	    zbroj = zbroj + i
	    print ('Nakon broja', i ,'zbroj je', zbroj)	

	a=0
	zbroj=0
	while a<21:
	    if a%2==0:
	        zbroj=zbroj+a
	        print(" Zbroj brojeva je",zbroj)
	    a=a+1	

VAŽNO:

U petlji WHILE , brojač može ići i unatrag
Primjer: Zadatak_2: Za ispis brojeva od 8 do 1
	a=8
	while a>=1:
	    print(a)
	    a=a-1

Zadaci za vježbu: Pokušajte zadatke riješiti pomoću ***petlje FOR i pomoću petlje WHILE***

Zadatak_1:

Ispiši samo parne brojeve do broja deset, ali i broj deset.

Zadatak_2:

Ispiši privih osam brojeva u obrnutom nizu (od 8 do 1)

a) Preuredi program pa ispiši samo parne brojeve u obrnutom nizu od 8 do 1
b) Preuredi program pa ispiši samo neparne brojeve u obrnutom nizu od 8 do 1

Zadatak_3:

a) Ispišite prvih 10 brojeva , ali bez broja 6
b) Ispišite prvih 20 brojeva ,ali bez broja 8 i broja 13 (sjetite se logičkih operatora)

Zadatak_4:

Napišite program koji ispisuje zbroj parnih brojeva od 1 do 20.

a) Napišite program koji ispisuje zbroj neparnih brojeva od 20 do 1

Zadatak_5:

Unesi neki broj preko tipkovnice(input) i ispiši sve brojeve do tog broja, ali i taj broj

Zadatak_6:

Unesi neki broj (input). Napravite ispis svih parnih brojeva do broja koji ste unijeli

a) Ispi zadatak samo za neparne brojeve

Zadatak_7:

Unesite neki broj(input) i napravite ispis svih brojeva do tog broja u obrnutom redoslijedu.

PETLJE – provjera znanja

Pitanje 1.

Koji od ponuđenih naredbi treba dodati na označeno mjesto da bi program tri puta ispisao riječi "Dobar dan" .

```
for ___________________:
 print("Dobar dan")
```

A. i in range(3)
B. i inrange(3)
C. i in 3
D. i = 3

Pitanje 2.

S kojom od ponuđenih linija koda , program NEĆE pet puta ispisati riječi "Dobar dan".

```
for ___________________:
 print("Dobar dan")
```

Odaberite točan odgovor:

A. i in range(1, 5)
B. i in range (5)
C. i in range(14, 19)

Pitanje 3. **

Što će biti rezultat sljedećeg programa, ako mu se na oba unosa unese broj 2 ?

```
for i in range(8):
        o = int(input("unesi broj opravdanih")
        n = int(input("unesi broj neopravdanih")
        u = o + n
        print(u)
```

A. 28 puta bit će ispisan broj 4.
B. 56 puta će se ispisat broj 2
C. 28 puta će biti ispisan broj 2.
D. Bit će ispisan broj 4

Pitanje 4.

Što će biti rezultat izvršenja sljedećeg programskog koda ?

```
for broj in range(1,5):
 print(broj)
```

A. Brojevi 1, 2, 3, 4 će biti ispisani.
B. Ispisat će se broj 5.
C. Brojevi 1, 2, 3, 4, 5 će biti ispisani.
D. Nijedan od ponuđenih odgovora nije točan.

Pitanje 5. *

Što će biti rezultat izvršenja sljedećeg programskog koda ?

```
for broj in range(5):
    print(broj)
```

A. Brojevi 1, 2, 3, 4 će biti ispisani.
B. Brojevi 0, 1, 2, 3, 4, 5 će biti ispisani.
C. Broj 5 će biti ispisan.
D. Nijedan od ponuđenih odgovora nije točan.

Pitanje 6. *

Što će biti rezultat izvršenja sljedećeg programa, ako mu se pri pokretanju za vrijednost a unese broj 9 , a za vrijednost b broj 15?

```
a = int(input("Unesi a"))
b = int(input("Unesi b"))
for i in range(a,b+1,8):
    print(i)
```

Pitanje 7. *

Što od ponuđenog bi ste unijeli u programski kod da se ispišu svi dvoznamenkasti brojevi koji su djeljivi sa 3,

```
for i in range(___________):
 print(i)
```

A. 10, 100, 3
B. 12, 100, 3
C. 12, 99, 3
D. 10, 99, 3

Pitanje 8. *

U igri skrivača brojimo do 200 ali po pet. Koju od ponuđenih linija koda bi ste iskoristili u programu za ispis brojeva do 200, uključujući i 200 s korakom od po 5.

```
for i in range(___________):
 print(i)
```

A. 0, 200, 5
B. 0, 210, 5
C. 5, 205, 5
D. 5, 200, 5

Pitanje 9. **

Što će biti rezultat izvršenja sljedećeg programskog koda ?

```
s = 0
for i in range(6):
 s = s + i
print(s)
```

Odgovor:

Pitanje 10. **

Što će biti rezultat izvršenja sljedećeg programskog koda?

```
p = 0
for i in range(4):
  p = p * i
print(p)
```

Odgovor:

STRINGOVI

Rad sa stringovima

Python poznaje nekoliko vrsta tipova podataka kao što su npr:

- **string** (niz znakova, piše se uvijek unutar navodnika)
- **int** (cijeli broj)
- **float** (decimalni broj)

Do sada smo uspješno radili sa tipovima podataka int i float

String tj.niz znakova definiramo **unutar navodnika.**

a="Pero"

b="123"

Stringove možemo **spajati.**

a="Pero"
b="123"

print(a+b)

Rješenje: Pero123

a="123"
b="456"

print(a+b)

Rješenje: 123456

String ne možemo spojiti s drugim tipom podataka npr. int.

a="123"
b=456

print(a+b)

Rješenje:

TypeError: Can"t convert "int" object to str implicitly

Ako želimo izvršiti istu operaciju nad dvije različite vrste podataka, potrebno ih je pretvoriti u istu vrstu podataka. String pretvaramo u cijeli broj(int) pomoću naredbe **int()**.

a="123"
b=456

print int(a)+b

Rješenje:
579

Svaki znak u stringu ima svoje mjesto koje nazivamo **indeks**. Mjesto tj.indeks definiramo pomoću **[] zagrade**.

a="Pero"

print a[0]

Rješenje:
P

Objašnjenje: Python mjesta broji od 0. U riječi Pero P ima indeks 0, e ima indeks 1, r ima indeks 2, o ima indeks 3.

P	**E**	**R**	**O**
[0]	[1]	[2]	[3]
[-4]	[-3]	[-2]	[-1]

Indeks možemo pregledavati i "unazad" , u tom slučaju zadnje slovo ima indeks **-1**, predzadnje -2 itd.

a="Pero"

print a[-1]

Rješenje: o

Neke osnovne naredbe za rad sa stringovima

Naredba	Objašnjenje	Primjer	Rješenje
len()	Duljina stringa	a="Pero" print len(a)	4
max()	Vraća najveći element stringa	a="56392" print max(a)	9
min()	Vraća najmanji element stringa	a="56392" print min(a)	2
a.replace()	Promijeni slova	a="Informatika" b=a.replace("a","o") print b	informotiko
a.count()	Prebroji određeno slovo	a="informatika" b=a.count("i") print b	2
a.endswith()	Provjeri s kojim znakom završava riječ		

Primjena gotovih funkcija u postupku programiranja

Primjer_1:

Prosječna ocjena

U listu su unesene ocijene iz nekoliko predmeta. Izračunajte prosječnu ocjenu.

ocjene = [5, 4, 5, 3, 5, 4, 4, 5]

prosjek = sum(ocjene) / len(ocjene)

print(prosjek)

Primjer_2:

Najniža ocjena

Ako znamo da su na listi tri posljednje ocjene iz prirodnih znanosti. Izračunajte najnižu ocjenu iz prirodnih znanosti.

ocjene = [5, 4, 5, 3, 5, 4, 4, 5]

ocjene_iz_prirodnih_znanosti = ocjene[-3:]

print(min(ocjene_iz_prirodnih_znanosti))

Razlika temperature

Primjer_3:

Vrijeme se često mijenja i u tijeku jednog tjedna izmjenjuju se topli i hladni dani.

Ako imamo unesene temperature za jedan tjedan, utvrdite razliku između najviše i najniže temperature.

temperatura = [17, 23, 12, 15, 19, 21, 25]

print(max(temperatura) - min(temperatura))

Primjer_4:

Visine djevojčica i dječaka u razredu

Poznate su visine djevojčica i dječaka u jednom razredu.

Napravite ispis , koliko ste visina unijeli tj. Koliko ima u tom razredu djevojčica i dječaka i izračunajte prosječnu visinu svih učenika.

```
visine_djevojcica = [165, 153, 155, 155, 157]
visine_djecaka = [170, 168, 173, 156, 172]
visine = visine_djevojcica + visine_djecaka
print(len(visine))
print(sum(visine) / len(visine))
```

Primjer_5.

Navedene su cijene tri proizvoda. Ako kupite sva tri proizvoda jedan od tih proizvod dobivate za 1 Kn. Koliko ćete platiti sve proizvode ?

```
cijene = [1420, 1799, 1569]
cijene_po_redu = sort(cijene)
cijene_po_redu[0] = 1
print(sum(cijene_po_redu))
```

Priimjer_6.

U listi imamo unesene cijene nekoliko proizvoda. Izračunajte sumu tri najjeftinija i tri najskuplja proizvoda.

```
cijene = [58.00, 104.95, 117.50, 11.95, 10.4, 37.95, 85.5]
sortirane_cijene = sort(cijene)
print("najjeftiniji" ,sum(sortirane_cijene[0:3]))
print("najskuplji" ,sum(sortirane_cijene[-3:]))
```

Primjeri zadataka s rješenjima:

1. Napiši program koji unosi ime i ispisuje poruku je li upisano ime muško ili žensko. Ako ime završava na a, ispisat će se poruka "Žensko ime", inače će se ispisati poruka "Muško ime.

```
a= input ("Upisi ime:")
if a[-1]=="a":
  print ("Zensko ime")
else:
   print ("Musko ime")
```

2. Napiši program koji dopušta unos dvije riječi i ispisuje poruku koja riječ je duža.

```
a= input ("Upisi prvu riječ: ")
b= input ("Upisi drugi riječ: ")
if len(a)>len(b):
   print ("Riječ",a,"je dulja.")
 else:
   print ("Rijec",b,"je dulja.")
```

3. Napiši program koji unosi rečenicu i ispisuje istu tu rečenicu u kojoj je svako slovo a promijenjeno u slovo o.

```
a= input ("Upisi recenicu: ")
b=a.replace("a","o")
print (b)
```

4. Napiši program koji unosi rečenicu u 3.licu jednine prezenta engleskog jezika i ispisuje ju u prošlom jeziku.

```
a= input ("Upisi recenicu: ")
b=a.replace("is","was")
print (b)
```

5. Napiši program koji unosi riječ i ispisuje poruku "Kalodont!" ako riječ završava na "ka". U protivnom će se ispisati poruka "Dalje".

```
a= input ("Upisi rijec: ")
 if a.endswith("ka"):
      print( "Kalodont!") else:
      print ("Dalje")
```

6. Napiši program koji unosi riječ i ispisuje poruku je li upisana riječ muškog, ženskog ili srednjeg roda. Ako riječ završava na o, ispisat će se poruka "srednji rod", ako riječ završava na "a" ispisat će se poruka "ženski rod". U svim ostalim slučajevima ispisat će se poruka "muški rod".

```
a= input (" Upisi riječ: ")
 if a[-1]=="o":
      print( "Srednji rod" )
elif a[-1]=="a"
      print( "Ženski rod" )
else:
      print ("Muški rod")
```

Zadaci za vježbu :

Zadatak_1:

Napiši program koji provjerava duljinu lozinke. Ako upisana lozinka ima manje od 8 znakova, ispisat će se poruka: Slaba lozinka. Ako lozinka ima 8 ili vise znakova ispisat će se poruka: Jaka lozinka.

Zadatak_2:

Napiši program koji unosi korisničko ime te provjerava je li korisničko ime upisano ili ne. Ukoliko je ispisat će pozdravnu poruku, a ako nije ispisat će poruku: "Niste upisali korisničko ime."

Podsjetnik kod definiranja unosa stringova (Znakova) koristimo samo naredbu input bez int ili float
Da li nešto je ili nije dovoljni je u funkciju if staviti prazno mjesto if=="", prazna mjesta su dva navodnika.

Zadatak_3:

Napiši program koji unosi tvoju najdražu riječ. Ispiši koliko slova ima ta riječ te poruku je li riječ duga ili kratka. Riječ je duga ako ima 7 ili više slova.

PRIMJERI ZADATAKA:

Pitanje 1.

Koji uvjet bi ste stavili u naredbu IF da bi ste dobili rješenje programa, ako je broj u listi ispisat će se da je u listi, ako nije ispisat će se da nije u listi

```
lista = [2,27,35,17]
a = int(input("Unesi jedan broj"))
if ( ):
  print("Uneseni broj se nalazi u listi")
else:
  print("Uneseni broj se ne nalazi u listi")
```

A. a in lista
B. len(a)>2
C. a exist in lista

Pitanje 2.*

Što će biti rezultat izvršenja sljedećeg koda?

```
lista = ["Edita", "Ana", "Petra"]
for ime in lista:
    print(ime)
```

A: Ispisat će se imena Edita, Ana i Petra.
B. Bit će ispisana riječ lista.
C. Ispisuju se brojevi 0,1 i 2.
D. Tri puta će se ispisati riječ “ime“

Pitanje 3. *

Što će biti rezultat izvršenja sljedećeg programskog koda?

```
ljubimci = ["pas", "mačka", "papagaj", "zec", "kornjača"]
for i in [1,3]:
   print(ljubimci[i])
```

A. Ispisat će se riječi mačka i zec.
B. Napisat će se riječi "pas" i "papagaj".
C. Napisat će se riječi "pas, mačka i papagaj".

Pitanje 4. *

Što će bit rezultat sljedećeg programskog koda?

```
ljubimci = ["pas", "macka", "papagaj", "zec", "kornjaca"]
for i in range(3):
    print(ljubimci[i])
```

A. Pisat će se riječi pas, mačka i papagaj.
B. Napisat će se riječi pas, mačka, papagaj i zec.
C. Papagaj će se ispisati.
D. Ispisat će se riječ zec.
E. Nijedan od ponuđenih odgovora nije točan.

Pitanje 5.

Što bi ste od ponuđenih naredbi umetnuli u naredbu for , da bi ispisali sve članove liste ljubimci.

```
for i in range(_____________):
  print(ljubimci[i])
```

A. 1, len (kućni ljubimci)
B. len (kućni ljubimci) - 1
C. ljubavnici
D. Nijedan od ponuđenih odgovora nije točan.

Pitanje 6.

Što će biti rezultat izvršenja sljedećeg programskog koda?

```
lista = [-3, 5, 16, 5, -2]
s = 0
for x in lista:
   s = s + x
print(s)
```

Pitanje 7.

Što će biti rezultat izvršenja sljedećeg programskog koda?

```
lista = [22, 20, 21, 24, 15, 19]
s = 0
for x in lista:
 if x % 2 == 0:
  s = s + x
print(s)
```

Odgovor:

Pitanje 8. **

Koja od navedenih tvrdnji je točna ?

```
r = []
slova = "abcdefghijklmnoprstuvz"
for x in slova:
  r[x] = 0
```

A. Stvorena je lista s malim slovima abecede.
B. Stvorena je lista s malim slovima abecede kojima je pridružena vrijednost 0
C. Program je pogrešno napisan i javit će pogrešku

Liste - Jednodimenzionalni niz

Liste su skupovi podataka (varijablā) koji predstavljaju jednu cjelinu. Elementi u listi odvojeni su zarezom, a mogu biti različitih tipova, za razliku od nekih drugih programskih jezika.

Također, lista se prepoznaje po uglatim zagradama unutar kojih se nalaze njezini elementi.

Primjer liste s različitim elementima u listi

lista = [1, “bicikl”, 3.14, “auto”]

Primjer stvaranja jedne liste:

lista=[3,6,9,12,23] – za liste obavezno koristimo uglate zagrade (desni Alt+f , desni Alt+g)

Za ispis i pristup elementima liste koristimo **indekse** :

	3	42	12	19	30	59
Index	0	1	2	3	4	5

Prvi element liste ima vrijednost 0 ,ako listu čitamo s lijeva na desno

Lista se može čitati i s desna prema lijevo , ali tada indeks ima negativnu vrijednost, prvi član liste s desne strane ima vrijednost -1.

Indeks	-6	-5	-4	--3	-2	-1

Ispis elemenata liste:

brojevi = [3, 42, 12, 19, 30, 59] print(brojevi[0])	Rezultat je 3
print(brojevi[1])	Rezultat 42

Brisanje pojedinog elementa s liste

brojevi = [59, 42, 30, 19, 12, 3, 199] print(brojevi[0])	Rezultat 59
brojevi.pop(0) print(brojevi)	Rezultat [42, 30, 19, 12, 3, 199]

Korištenje više elemenata liste odjednom

Pravilo : [početni_index:završni_index+1]

lista= ['a', 'b', 'c', 'd']	**Rezultat ['b']**
print(lista[1:2])	

Određeni niz elemenata možemo zamijeniti drugim elementom

lista= ['a', 'b', 'c', 'd']	Zamjenjuje ['a', 'b'] sa ['z']
lista[0:2] = 'z'	
print(lista)	['z', 'c', 'd']

Važno je znati

lista = ['a', 'b', 'c', 'd'] print(lista[:])	['a', 'b', 'c', 'd']

Ukoliko ne definiramo niz, već stavimo samo [:], ispisat će se svi elementi liste

Članove liste možemo ispisivati i pomoću petlje:

lista=	[	3	6	9	12	23	]

for i in range(len(lista)): print(lista[i])	funkcija len(lista) daje nam brojčanu vrijednost koliko imamo članova liste. Pri ispisu liste obavezno moramo navesti naziv liste i u uglate zagrade staviti naziv brojača pomoću kojeg će se petlja kretati po listi.

Primjer: Programski primjer korištenja negativnog indeksa

Napiši program koji unosi ime i ispisuje poruku je li upisano ime muško ili žensko. Ako ime završava na a, ispisat će se poruka 'Žensko ime', inače će se ispisati poruka 'Muško ime.	a= input ('Upisi ime: ')
	if a[-1]=='a':
	  print 'Zensko ime'
	else:
	  print 'Musko ime'

Provjera da li je neki element u listi ili ne

Sjetimo se četvrtog oblika petlje **for** i korištenje naredbe **in ili not in** umjesto naredbe **range**.

Primjer:

lista=	[	3	6	9	12	23	]
d							

for d in lista:
  print("Elementi liste",d)
 	
rezultat: 3 ,6, 9, 12, 23
	Zadatak: Napišite riječ "SLAVICA" i ispišite svako slovo jedno ispod drugoga.
	
	x=" SLAVICA"
	for k in x:
	  print(k)
	rezultat: ispisat će slovo po slovo riječi SLAVICA

Umetanje elemenata liste na kraj liste

niz=[3,5,7,8,12,13] niz.append(15) print(a)	Metoda append dodaje broj 15 na kraj liste imena niz. Pa će naša lista izgledati ovako niz=[3,5,7,8,12,13,15]

Liste uz samo dodavanje elemenata na kraj liste imaju još mnogo mogućnosti, a najčešće su sljedeće:

Naredba	Objašnjenje	Primjer	Rješenje
max(a)	Vraća najveći element u listi	a=[3,5,7,8,3,2,4] print max(a)	8
min(a)	Vraća najmanji element u listi	a=[3,5,7,8,3,2,4] print min(a)	2
sum(a)	Zbraja sve elemente u listi	a=[3,5,7,8,3,2,4] print sum(a)	32
a.sort()	Sortira elemente u listi	a=[3,5,7,8,3,2,4] a.sort() print (a)	2,3,3,4,5,7,8
del (a)	Briše element liste	a=[3,5,7,8,3,2,4] del a[0] print (a)	5,7,8,3,2,4
a.append()	Dodaje element na kraj liste	a=[3,5,7,8,3,2,4] a.append(6) print (a)	3,5,7,8,3,2,4,6
len(a)	Brojčano prikazuje duljinu liste	a=[3,5,7,8,3,2,4]	7

Primjer s objašnjenjem – stvaranje prazne liste i unos elemenata u listu

Želimo napraviti program za unošenje nekoliko imena u listu, korisnik sam bira koliko elemenata želi unijeti.

Prvo trebamo jednu varijablu u koju će se spremiti unos od strane korisnika s pomoću koje ćemo znati koliko imena on želi unijeti te praznu listu u koju ćemo unositi vrijednosti:

unos = int(input(“Koliko imena želite unijeti? “))
lista = []

Nakon toga možemo krenuti na for petlju, gdje ćemo popunjavati listu:

```
for i in range(0,unos):
    x = input(“Unesite ime: “)
    lista.append(x)
```

Nakon što smo u for petlji napravili varijablu x u koju se sprema ime i nakon toga se dodaje na kraj liste, potrebno je nakon izvršenja petlje ispisati cijelu listu, a nakon toga pitati korisnika koje ime želi da se ponovo ispiše.

Imena nećemo ispisivati svako posebno, već ćemo ispisati listu, a to radimo ovako:

print(lista)
broj = int(input(“Unesite broj za ispis imena “))

Sada nam samo preostaje ispisati ime iz liste. Međutim, ovdje se treba sjetiti da elementi počinju od 0, a ne od 1, stoga se vrijednost mora umanjiti za 1:

print(lista[broj-1])

Na ovaj način napisali smo program, a kako bi to trebalo izgledati pogledajte na sljedećoj slici:

```
unos = int(input("Koliko imena želite unijeti? "))
lista = []
for i in range(0,unos):
        x = input("Unesite ime: ")
        lista.append(x)
print(lista)
broj = int(input("Unesite broj za ispis imena "))
print(lista[broj-1])
```

rješenje: unesimo imena [Ana, Marica, Slavica]

ako odaberem da želimo drugog člana liste , što će se ispisati ?

rješenje: Marica , zašto ?– zbog **print(lista[broj-1])** **tj. zbog -1.**

Primjeri zadataka:

Zadatak_1:

Napiši program koji provjerava duljinu lozinke. Ako upisana lozinka ima manje od 8 znakova, ispisat će se poruka: Slaba lozinka. Ako lozinka ima 8 ili vise znakova ispisat će se poruka: Jaka lozinka.	**a=input("Upiši lozinku:")
	if len(a)<8:
	    print ("Slaba lozinka")
	else:
	    print ("Jaka lozinka")
	
	npr:
	
	Upiši lozinku: luka123
	Slaba lozinka**

Zadatak_2:

Napiši program koji za upisanu riječ ispisuje samo samoglasnike.	**a=input("Upiši riječ:)
	for i in a:
	    if i in "aeiouAEIOU":
	        print (i)
	
	npr:
	Upiši riječ: informatika
	i o a i a**

Zadatak_3:

Stvorimo niz točno određene duljine (maksimalno 5 mjesta) i unesimo podatke preko tipkovnice
niz=[0]*5 , *5 -koristimo da bi smo listu ograničili na maxsimalno pet mjesta za unos podataka u listu.
	niz=[0]*5
	for i in range(0,5):
	    niz[i]=int(input("Unesite broj"))
	for i in range(0,5):
	        print(niz[i])

Pojašnjenje :

niz=[0]*5
for i in range(0,5):
    niz[i]=int(input("Unesite broj"))
for i in range(0,5):
        print(niz[i])
	– definiramo niz za unos 5 elemenata
	- Za unos elemenata u niz moramo definirati varijablu **niz[i].**
	- Ispis svih elemenata niza

Zadatak_4:

Kreirajmo niz od maksimalno 10 elemenata koji će se sam popuniti i zbrojite sve parne članove niza.
 	
Ispravite pogreške u programu
	niz=[0]*10
	zbroj=0
	#automatsko popunjavanje niza
	for i in range(0,10):
	    niz[i]=niz[i]+1
	#zbroj parnih brojeva u nizu
	for i in range(0,10):
	       if niz[i]%2==0:
	           zbroj=zbroj +niz[i]
	#ispis niza
	for i in range(0,10):
	        print(niz[i])
	print(" zbroj parnih brojeva",zbroj)

Primjeri zadataka:

Primjer_1:

Unesite 6 elemenata u listu i ispišite sve parne brojeve u listi.

<table>
<tr>
<td>

```
niz=[0]*6 #deklaracija niza
#unos podataka u niz
for i in range (0,6):
   niz[i]=int(input("unesite  broj: " ))

for i in range (1,6,2):
   if niz[i]%2==0:
      print(niz[i])
```

</td>
<td>

Zadaci za vježbu :

a) **Preuredite program tako da se ispišu svi neparni brojevi u nizu.**
b) **Ispišite sve brojeve niza djeljive sa 3.**
c) **Ispišite sve brojeve djeljive sa 3 i 4.**
d) **Ispišite sve brojeve djeljive sa 3 ili 4.**

</td>
</tr>
</table>

Primjer_2: Ispis i unos podataka iz liste u novu listu , korištenjem funkcije append

<table>
<tr>
<td>

```
lista=[3,6,9,12,15]
for index in range(len(lista)):
        print("lista[",index,"]==",lista[index])
```

</td>
<td>

Uređeni ispis podataka iz liste

```
lista[ 0 ]== 3
lista[ 1 ]== 6
lista[ 2 ]== 9
lista[ 3 ]== 12
lista[ 4 ]== 15
```

</td>
</tr>
</table>

Zadatak:_1 :

<table>
<tr>
<td>

```
lista=[3,6,9,12,15]
nova_lista=[]
for i in lista:
   if i%2==0:
      nova_lista.append(i)

print(nova_lista)
```

</td>
<td>

Ispis parnih brojeva iz liste (lista) i njihovo spremanje u novu listu (nova_lista)

</td>
</tr>
</table>

Zadatak_2:

<table>
<tr>
<td>

```
lista=[]
broj=1
while broj>0:
    broj=int(input("Unesi broj:  "))
    lista.append(broj)
lista.sort()

parni,neparni=[],[]
for br in lista:
   if br%2==0:
      parni.append(br)
   else:
      neparni.append(br)
print("lista: ",lista)
print("Parni brojevi: ",parni)
print("Neparni brojevi: ",neparni)
```

</td>
<td>

Kreirat ćemo praznu listu u koju ćemo unositi brojeve sve dotle dok je broj veći od 0. Kada unesemo negativan broj prestat ćemo s unosom i kreirat ćemo dvije liste:

1) **za parne brojeve**
2) **za neparne brojeve**

Zadaci:

a) **Ispišite sve brojeve niza djeljive sa 3 i rezultat pohranite u novu listu.**
b) **Ispišite sve brojeve djeljive sa 3 i 4.i rezultat pohranite u novu listu**
c) **Ispišite sve brojeve djeljive sa 3 ili 4. i rezultat pohranite u novu listu**

</td>
</tr>
</table>

PRIMJERI ZADATAKA ZA PROVJERU ZNANJA

Pitanje 1.

Kako bi ste definirali listu koja sadrži brojeve 3, -5, 4, -1 ?

a) lista = [3, -5, 4, -1]
b) lista = (3, -5, 4, -1)
c) lista = {3, -5, 4, -1}

Pitanje 2.

Kako bi ste definirali listu koja sadrži brojeve -1, 63 i ime Petar?

a) list = [-1, 63, "Petar"]
b) list = [-1, 63, Petar]
c) list = {-1, 63, "Petar“}
d) Nije moguće istovremeno definirati list s brojevima i tekstom.

Pitanje 3.

Koja je vrijednost šestog elementa liste?

lista = [76, 187, 163, 193, 141, 103, 163]

Odgovor:

Pitanje 4.

Čemu je jednak indeks šestog elementa liste?

lista = [103, 154, 71, 120, 189, 185, 71]

a) 5
b) 6
c) 185
d) 71

Pitanje 5.

Što će biti rezultat izvršenja sljedećeg koda?

```
lista = [4, -4, 1, 3]
print( lista[ 2 ] )
```

a) 1
b) -4
c) [4, -4]
d) [-4, 3]

Pitanje 6. *

Što će biti rezultat izvršenja sljedećeg koda?

```
lista = [10, 10, 9, 12, 15]
lista[ -1 ] = 0
print( lista )
```

a) [10, 10, 9, 12, 0]
b) [10, 10, 9, 0, 15]
c) [-1, 10, 9, 12, 15]
d) Popis će ostati nepromijenjen. Element s indeksom -1 ne postoji, tako da se naredba neće izvršiti.

Pitanje 7.

Što će biti rezultat izvršenja sljedećeg koda?

```
lista = [-3, -1, 1, 1]
print( len( lista ) )
```

Odgovor:

Pitanje 8. *

Što će biti rezultat izvršenja sljedećeg koda?

```
lista1=[4, -2, "pero", "1"]
print( len( lista1 ) )
```

a) 2
b) 3
c) 4

Pitanje 9. *

Naveden je popis cijena sladoleda u jednoj trgovini. Kojom naredbom možete dobiti najnižu cijenu sladoleda?

```
lista = [87, 74, 25, 26, 46]
```

a) min (popis)
b) lista [-1]
c) popis [0]
d) popis [1]

Pitanje 10. *

Što će biti rezultat izvršenja sljedećeg koda?

```
lista=[4, 1, 3, -5]
print( sum( lista ) / len( lista ) )
```

a) 0
b) 0.75
c) 3

Pitanje 11.

Kojom ugrađenom funkcijom u pythonu možemo sortirati list ?

a) sorted
b) min
c) max
d) sort

Pitanje 12. *

Što će biti rezultat izvršenja sljedećeg koda?

```
voce = [50, 40, 45, 80, 65]
cijena = sort( voce )
print( cijena )
```

a) [50, 40, 45, 80, 65]
b) [40, 45, 50, 65, 80]
c) [2, 0, 1, 4, 3]

Pitanje 13. **

Što će biti rezultat izvršenja sljedećeg koda?

```
povrce = [73, 162, 154, 142, 172, 75, 80]
cijena = sort(povrce)
print(sum( cijena[0 : 4] ) )
```

a) 531
b) 710
c) 370

Pitanje 14. **

Što će biti rezultat izvršenja sljedećeg koda?

```
sladoled = [77, 140, 174, 54, 59, 194, 105]
cijena = sort(sladoled)
print( cijena[-3 : ] )
```

a) Cijene tri najskuplje sladoleda.
b) Cijene četiri najskuplja sladoleda.
c) Cijena najjeftinijeg sladoleda.
d) Posljednja naredba nije ispravno napisana, pa će Python dati poruku o pogrešci

Pitanje 15. **

Što će biti rezultat izvršenja sljedećeg koda?

sladoled = [183, 105, 106, 136, 151, 166, 124]
print(sort(sladoled)[0 : 4])

a) [105, 106, 124, 136]
b) [105, 106, 136, 183]
c) [105, 106, 124, 136, 151]
d) [105, 106, 136, 151, 183]
e) Posljednja naredba je pogrešno napisana, pa će Python dati poruku o pogrešci .

Pitanje 16. **

Što će biti rezultat izvršenja sljedećeg koda?

sladoled = [171, 148, 50, 50, 126, 142, 178]
print(sort(sladoled[0 : 4]))

a) [50, 50, 126, 142]
b) [50, 50, 148, 171]
c) [50, 50, 126, 142, 148]
d) [50, 50, 126, 148, 171]

Pitanje 17. *

lista = ['Marko Marković', 'Ivan Ivić', 'Petar Petić', 'Ivan Tot', 'Mario Car', 'Ivan Nađ']

Definirali smo listu s popisom učenika kako su navedeni u dnevniku, kojom naredbom biste dobili broj u dnevniku na kojem se nalazi Ivan Tot.

a) lista.index ("Ivan Tot") + 1
b) lista.find ("Ivan Tot")
c) lista.index ("Ivan Tot")

Pitanje 18. *

Što će biti rezultat izvođenja sljedećeg programa?

l= ['Marko Marković', 'Ivan Ivić', 'Petar Petić', 'Ivan Tot', 'Mario Car', 'Ivan Nađ']
k = l.index("Mario Car")
print(l[k-1])

a) Marko Marković
b) Ivan Nađ
c) Ivan Ivić

Pitanje 19. *

Što će biti rezultat izvođenja sljedećeg programa?

```
l= ['Marko Marković', 'Ivan Ivić', 'Petar Petić', 'Ivan Tot', 'Mario Car', 'Ivan Nađ']
k = len(l)
print( l[ k ] )
```

a) Ivan Nađ
b) Ivan Ivić
c) Pojavit će se pogreška jer smo pogrešno napisali programski kod.

Pitanje 20. *

Neka je mjerena najviša dnevna temperatura tokom 10 dana u mjesecu siječnju. Koliko puta je dnevna temperatura bila jednaka 0 ?

```
l = [14.2, 9.0, 10.0, 18.6, -2.9, 3.6, 17.7]
___________
print( k )
```

Koju od sljedećih naredbi bi koristili za točno rješenje zadatka ?

a) k = l.count (0)
b) k = l.count ("0")
c) k = broj (l, 0.0)
d) Nijedan od ponuđenih odgovora nije točan.

Pitanje 21. **

Što će biti rješenje ovog programskog koda?

```
l = [2, 5, 8, 9, 13, 1, 9]
k = 3
l[k] = l[0]
l[0] = k
print( l )
```

a) [3, 5, 8, 2, 13, 1, 9]
b) [3, 5, 2, 9, 13, 1, 9]
c) [2, 5, 8, 9, 13, 1, 9]

Pitanje 22. *

Što će biti rješenje ovog programskog koda?

```
l = [13, 11, 18]
k = [13, 16, 15]
n = l + k
print( n )
```

a) [13, 11, 18, 13, 16, 15]
b) [26, 27, 33]
c) 86
d) Treći redak koda nije ispravna Python naredba. Ispisat će se poruka o grešci

Python – Funkcije

Kod kompliciranijih zadataka imamo relativno puno redaka koda, često i dijelova koji se ponavljaju, ali su nam potrebni pri samoj izvedbi programa. Kako bismo izbjegli ponavljanje i ponovno pisanje istog ili sličnog koda, koristimo funkcije.

Funkcije su grupirani dio koda koji izvršava određenu radnju. Primjerice, imamo program gdje na različitim dijelovima zbrajamo i ispisujemo vrijednost. Ako to želimo raditi bez funkcija, nekoliko puta moramo pisati isti kod. S druge strane, uporabom funkcije, to nam neće biti potrebno.

Funkcije mogu biti ugrađene u programski jezik, a mogu ih pisati i korisnici. S ugrađenim funkcijama smo se sreli: **print(), len(), del()..,** dok funkcije koje piše korisnik tek trebamo objasniti.

Kod već ugrađenih funkcija vidimo da se samo imenom funkcije i korištenjem argumenata (tj. vrijednosti unutar zagrada) lako može doći do konačne vrijednosti koja nam je potrebna. Tako ne pišemo dodatne redove koda, nego uz pomoć funkcije jednostavno izvršimo određeni zadatak.

Na isti način djeluju i funkcije koje napiše korisnik. Ono što čini funkciju je ključna riječ **def**, nakon toga ime funkcije, pa zagrade unutar kojih možemo upisati argumente te dvotočka i uvučeni blok naredbi nakon toga:

```
def ime_funkcije():
// blok naredbi
```

Na ovaj način je definirana funkcija, dok se ona poziva na sljedeći način:

ime_funkcije()

Zapravo, sve je isto kod poziva funkcije kao i kod ugrađenih funkcija programskog jezika. Da sve to ne bi bila samo teorija, riješit ćemo jedan primjer uz pomoć funkcija. Zadatak koji ćemo riješiti je zbrajanje brojeva uz pomoć funkcija.

Prvo nam je potrebno definirati funkciju. Nazvat ćemo ju **zbrajanje**. Sljedeće što nam treba je unutar tijela funkcije napisati kod koji će tražiti unos dva broja te ispisati njihov zbroj. To će izgledati ovako:

```
def zbrajanje():
    a=int(input("Unesite prvi broj:=    "))
    b=int(input("Unesite drugi broj: =  "))
    print("Zbroj brojeva",a,"i",b,"je",a+b)
```

Nakon što smo definirali funkciju, potrebno ju je pozvati kako bi se ona mogla izvršiti. Pošto ćemo napraviti program koji zbraja brojeve **pet puta**, koristit ćemo **for** petlju:

```
for i in range(1,6):
    zbrajanje()
    print("")
```

Sada smo gotovi s izradom i pozivanjem funkcije te možemo pokrenuti program:

Kao što vidite, korištenje funkcija nam je skratilo vrijeme pisanja programa jer smo za zbrajanje i ispis samo jednom pisali kod i nije bilo potrebno kasnije to sve pisati ponovo.

Druga pozitivna stvar je povećanje čitkosti koda. Na vrhu nam se nalazi funkcija u kojoj piše što se radi, dok se kasnije ona samo poziva da izvrši određeni zadatak.

No, treba napomenuti jednu stvar. **Funkcija mora biti definirana prije pozivanja iste**, jer će program inače javiti pogrešku (u našem primjeru funkcija je napisana odmah na početku, a kasnije je tek pozivana).

Sljedeća stvar koju ćemo spomenuti su argumenti. Njih smo već sreli, i to kod gotovo svih ugrađenih funkcija koje smo koristili.

Argumenti su vrijednosti koje prosljeđujemo funkciji. Prosljeđujemo ih tako da unutar zagrade napišemo vrijednost ili ime varijable, liste i sl. koja sadržava određenu vrijednost.

print("Pozdrav")

U ovom primjeru vidimo da funkcija print() ima argument "Pozdrav". Pošto je to funkcija ispisa, ona ispisuje upravo vrijednost tog argumenta.

Kod funkcija definiranih od strane korisnika, korisnik sam određuje koliko će argumenata biti potrebno za izvođenje određene funkcije. Za primjer ćemo uzeti funkciju koja množi dva broja i ispisuje vrijednost. Prvo ćemo definirati dvije varijable, gdje će se u svaku spremiti vrijednost koju korisnik unese. Nakon toga će se funkcija pozivati, a u zagradi će se nalaziti proslijeđeni argumenti (brojevi koje je unio korisnik).

```
def zbrajanje(a,b):
    print("Zbroj brojeva",a,"i",b,"je",a+b)


x=int(input("Unesite prvi broj =  "))
y=int(input("Unesite drugi broj =  "))
zbrajanje(x,y)
```

Ono što može biti zbunjujuće su imena argumenata pri definiranju i pozivu funkcije. Kod definiranja funkcije, imena argumenata možemo odabrati svojevoljno i ne moraju imati ista imena koja će kasnije u ostatku programa biti korištena. **Međutim, u samoj definiciji funkcije, imena argumenata unutar zagrada i u tijelu funkcije moraju biti ista.**

Ovo funkcionira tako da se pri pozivu funkcije stvore tipovi podataka (varijable, liste…) iste kao i proslijeđeni tip podataka, a vrijednosti se preslikaju u njih. Kada funkcija završi, ti se pomoćni tipovi podataka brišu iz memorije.

Upravo iz razloga da se pomoćni podaci korišteni u funkcijama brišu, odnosno, nakon izvršavanja funkcije da se oslobodi korištena memorija, postoji mogućnost da funkcija vrati vrijednost.

Zašto?

Čisto iz praktičnih razloga, odnosno, ako neku vrijednost iz funkcije trebamo koristiti u daljnjem dijelu programa bilo bi korisno da tu vrijednost sačuvamo.

<u>Možemo vrijednost poslati u funkciju, i funkcija tu vrijednost može vrati nazad.</u>

Primjer: Unesite dva boja i zbrojite ih.

<table>
<tr>
<td rowspan="2">

```
def brojevi(x, y):
    return x + y


#glavni program

broj = brojevi(1,2)
print(broj)
```

</td>
<td>

```
Rezultat ovog primjera će biti 3.

U ovom primjeru mi smo željeli
zbrojiti brojeve (1 i 2). Unijeli
smo ih izravno u funkciju.

Kad bi mi htjeli unositi brojeve,
glavni program bi izgledao ovako:
```

</td>
</tr>
<tr>
<td>

```
def brojevi(x, y):
    return x + y


#glavni program

a=int(input(" Unesi prvi broj:  "))
b=int(input("Unesi drugi broj:  "))
broj = brojevi(a,b)
print(broj)
```

</td>
</tr>
</table>

Kako bismo to postigli koristimo naredbu **return**. Uzet ćemo za primjer zadatak gdje izračunavamo aritmetičku vrijednost liste.

```
def aritmeticka_sredina(lista):
    suma=0
    for a in lista:
        suma=suma+a
        rezultat=suma/len(lista)
        return rezultat


lista=[3,5,7,5,8,2,1]
rezultat=aritmeticka_sredina(lista)
print("Aritmetička sredina iznosi",rezultat)
```

Na početku smo definirali funkciju koja prolazi kroz listu i računa aritmetičku vrijednost.

Nakon toga, definirali smo varijablu rezultat u koju ćemo spremiti vraćenu vrijednost.

Naravno, kako bi u varijablu spremili vraćenu vrijednost, koristimo znak pridruživanja. Ovo bi mogao biti primjer dijela programa gdje računamo prosjek nekog učenika koji kasnije koristimo u neke druge svrhe.

Primjeri funkcija:

Zadatak_1: Napišite funkciju koja od vas traži da 5 puta unesete neki broj

def unos(): a=int(input("Unesite jedan broj"))	**Ovaj dio programa je pod funkcijom** unos()
print("poziv funkcije") for i in range(1,5): unos()	**Ovo je dio glavnog programa , on nije pod funkcijom (vidite po uvlaci varijable a).**

Zadatak_2:

Napišite funkciju za zbrajanje dva broja , program se poziva 3 puta.

def zbroj():
    a=int(input("Unesite jedan broj =   "))
    b=int(input("Unesite drugi broj =   "))
    c=a+b
    print("zbroj brojeva",a,"i",b,"je",a+b)
	Cijeli dio ovog programskog koda spada pod funkciju

# glavni program for i in range(1,4): zbroj()	Glavni dio programa u kojem pozivamo funkciju

Svi primjeri do sada su primjeri funkcije bez argumenata . Idealne su za programa koji imaju mogućnost višestrukog odabira tj. izbornici npr. Kalkulator s više operacija

Funkcije s argumentima

Primjer_1:

def mnozenje(a,b): print("Umnožak brojeva",a,"i",b,"je",a+b) # glavni program b1=int(input("Unesi prvi broj ")) b2=int(input(unesi drugi broj ")) #poziv funkcije s argumentima mnozenje(b1,b2)	Argumenti u funkciji nam omogućuju unos vrijednosti iz glavnog programa . Imena argumenata koja su definirana u funkciji ne moraju biti ista. Bitno je da broj argumenata u funkciji bude jednak broju argomenata koji ulaze u funkciju.

Primjer_2:

Napišite program za unos nekog broja i provjeru da li je taj broj paran ili negativan.

Definirat ćemo dvije funkcije, jednu za provjeru da li je broj paran, a drugu da li je broj neparan.

<table>
<tr>
<td>

```
def paran(n):
    if n%2==0:
        print("broj je paran",n)
    else:
        print("Broj nije paran",n)

def negativan(n):
    if n<0:
        print("broj je negativan",n)
    else:
        print("Broj je pozitivan",n)

#glavni program

unos=int(input("Unesite neki broj  "))
paran(unos)
negativan(unos)
```

</td>
<td>

Primjer korištenja više funkcija u okviru jednog programa. To predstavlja ozbiljnije programiranje.

Pogledajte kod:! – funkciju paran koristimo za provjeru parnosti broja.

Funkciju negativan koristimo za provjeru da li je broj pozitivan ili ne.

Glavni program omogućuje unos broja i poziva funkcije da provjere u koju funkciju taj broj pripada

<u>Zadatak</u>: nadopunite program tako da tri puta unesete brojeve i za svaki broj da se provjeri njegov status.

</td>
</tr>
</table>

Možemo vrijednost poslati u funkciju, i funkcija tu vrijednost može vrati nazad.
RETURN

Primjer_1:

<table>
<tr>
<td>

```
def zbroj(n):
    s=0
    for i in range(1,n+1):
        s=s+i
    return s

def unos():
    n=int(input("Unesi neki broj  "))
    print(zbroj(n))

unos()
```

</td>
<td>

Unesimo neki broj npr: 3, rezultat će nam biti 6 kako??

Broj 3 nam ulazi u funkciju zbroj i ulazi u petlju for.

Prolazi kroz petlju

1. s=0+1
2. s=1+2
3. s=3+3
4. return s pamti vrijednost 6 i vraća vrijednost funkcije nazad , a print(zbroj(n)) ispisuje vrijednost koju je primila od funkcije zbog naredbe RETURN.

</td>
</tr>
</table>

Primjer_2:

Zbrajanje dva broja : u funkciji zbroj , zbrajanje je izvršeno u naredbi return a+b	<pre>def zbroj(a,b):
	    return a+b
	
	def unos():
	    a=int(input("Unesi prvi broj "))
	    b=int(input("Unesi drugi broj "))
	    print(zbroj(a,b))
	unos()</pre>

Primjer_2a:

Prethodni zadatak ali sada funkcija zbroj vraća pomoću return rezultat zbrajanja.
 	
Obratite pažnju , u parametrima unosa koristili smo varijable f i g , a u funkciji zbroj koristili smo a i b.
 	
Zadatak: *dopunite program taka da zbraja tri broja !!*
	<pre>def zbroj(a,b):
	    c=a+b
	    return c
	
	def unos():
	    f=int(input("Unesi prvi broj "))
	    g=int(input("Unesi drugi broj "))
	    print(zbroj(f,g))
	
	unos()</pre>

Primjer_3:

Napravi program koji pomoću funkcije ispisuje broj pojavljivanja slova A u nekom tekstu koji sami unosimo.
 	
Nadam se da ste razumjeli naredbu return.
 	
Hvala!!
	<pre>def zbroj(s):
	   d=len(s)
	   br=0
	   for i in range(d):
	      if s[i]=="A" or s[i]=="a":
	         br=br+1
	   return br
	#glavni program
	tekst=input("Unesi tekst: ")
	print("Broj pojavljivanja slova A: ", zbroj(tekst))</pre>

Funkcija unosa i ispisa liste

``def unos(niz):
    for i in range(0,5):
        niz[i]=int(input("Unesi elemnte niza "))
 	
def ispis(niz):
    for i in range(0,5):
        print(niz[i])
 	
#glavni program
 	
niz=[0]*5
unos(niz)
ispis(niz)```
	Primijenimo stečena znanja o funkcijama na listama za njihovo stvaranje, ispis i razne operacija nad elementima liste.

Složeni primjer sa izvršavanjem programa sa mogućnošću odabira operacije

<pre>def unos(niz):
    for i in range(0,5):
        niz[i]=int(input("Unesi elemnte niza "))
 	
def ispis(niz):
    for i in range(0,5):
        print(niz[i])
 	
def obrnuto(niz):
    for i in range(4,-1,-1):
        print(niz[i])
 	
def parni(niz):
    for i in range(1,5,2):
        print(niz[i])
 	
#glavni dio programa i poziv funkcije
niz=[0]*5
suma=0
 	
izbor=0
while 1:
    print("odaberite")
    print("1. Unos elemenata niza")
    print("2. Ispis elemenata niza")
    print("3. Obrnuti ispis elemenata niza")
    print("4. Ispis parnih elemenata niza")
    print("5. IZAZ IZ PROGRAMA")
    izbor=int(input(" Odaberite što želite"))
    if izbor==1:
        unos(niz)
    elif izbor==2:
        ispis(niz)
    elif izbor==3:
        obrnuto(niz)
    elif izbor==4:
        parni(niz)
    elif izbor==5:
        break</pre>
	Pogledajte petlju while i način njene primjene pri stvaranju složenijih programa s mogućnošću izbora.

PRIMJERI ZA PROVJERU ZNANJA

Pitanje 1.

Što će biti rezultat izvršavanja sljedećeg koda?

```
def opseg(a):
    return 4 * a

print(opseg(6))
```

Odgovor:

Pitanje 2.

Koju od ponuđenih linija koda treba dodati na označeno mjesto kako bi se ispravno definirala funkcija koja izračunava kvadrat danog broja?

```
def kvadrat(a):
```

A. return a * a
B. return * 2
C. a * a
D. return kvadrat

Pitanje 3.

Što će biti rezultat izvršavanja sljedećeg koda?

```
def f(a):
    return - 3 * a

print(f(0) - f(-1))
```

Odgovor:

Pitanje 4. *

Što će biti rezultat izvršavanja sljedećeg koda?

```
def f(a):
    return 2 * a + 3

print(f(-2) - f(f(2)))
```

Odgovor:

Pitanje 5. *

Koju vrijednost treba imati varijabla m , da bi rezultat funkcije bio 15 ?

```
def f(a):
    if a % 5 == 0:
        return 2 * a
    else:
        return a + 1

m = int(input("unesi cijeli broj"))
print(f(m))
```

Pitanje 6. *

Editi je trebalo neko vrijeme da pročita knjigu, vrijeme je izraženo u minutama. Ispišite to vrijeme u satima i minutama. Koju od ponuđenih lija koda ćete uzeti da uspješno riješite zadatak

```
def vrijeme(a):
    s = a // 60
    m = a % 60
 ______________

x = int(input("Unesi koliko minuta je Edita čitala knjigu"))
(s,m) = f(x)
m = int(input("unesi cijeli broj"))
print(s, m)
```

A. return s, return m
B. return s, m
C. (s, m)
D. return (s, m)

Pitanje 7. **

Što će biti rezultat sljedećeg programa?

```
def f(l,n):
    return l + n

print(f(11,22)," ",f("11","22"))
```

A. 33 "33"
B. 33 "1122"
C. 1122 "1122"
D. 33 33
E. Python okruženje prijavit će pogrešku tijekom izvršavanja zadanog programa.

Pitanje 8. **

Što će biti rezultat sljedećeg programa?

```
def f(l,n):
   return l * n

print(f(2,"3"))
```

A. 6
B. "222"
C. 33
D. Python okruženje prijavit će pogrešku tijekom izvršavanja zadanog programa.
E. Nijedan od ponuđenih odgovora nije točan.

Pitanje 9. **

Dana je funkcija koja izračunava kvadrat zadanog broja.

```
def f(a):
    return a * a
```

Navedite ispravnu oznaku linije koda u kojoj se koristi funkcija koja ispisuje kvadrate svih brojeva od 4 do 10.

```
1. print([f(x) for x in range(4,10)])
2. print([f(a) for x in range(4,10)])
3. print([f(x) for x in range(4,11)])
4. print([f(a) for x in range(4,11)])
```

Odgovor:

Pitanje 10. *

Navedi oznaku funkcije koja će za dati dvoznamenkasti broj, vraća zbroj znamenaka jedinica i desetica.

```
1. def dvocifren(a):
2.         d = a // 10
3.         j = a % 10
4.         return sum(j, d)


5. def dvocifren(a):
6.         d = a // 10
7.         j = a % 10
8.         return (j, d)


9. def dvocifren(a):
10.        d = a // 10
11.        j = a % 10
12.        return j + d
```

Odgovor:

Pitanje 11. **

Zadana je funkcija kojom se izračunava opseg trokuta

```
def f(a, b, c):
 return a + b + c
```

Navedite programsku oznaku koju koristi zadanu funkciju za ispis opsega nekoliko trokuta čije su veličine stranica navedene u listi trokuti.

```
trokuti = [(3, 4, 5), (5, 12, 13), (7, 24, 25)]
for trokut in trokuti:
 print(opseg(*trokut))

trokuti = [(3, 4, 5), (5, 12, 13), (7, 24, 25)]
for trokut in trokuti:
 print(opseg(trokut))

trokuti = [(3, 4, 5), (5, 12, 13), (7, 24, 25)] for i in range(len(trokuti)):
        print(opseg(trokuti[i]))

```

Odgovor:

Pitanje 12. **

Dat je jedan dio programskog koda.

```
pravokutnik = [(3, 9), (4, 9), (5, 10)]
for p in pravokutnik:
    print(površina(p))
```

Koja od sljedećih definicija funkcije može dati površina kako bi navedeni kôd radio ispravno?

```
def povrsina(a):
    return a * a

def povrsina(a):
     return a[0]*a[1]

def povrsina(a):
    return a(0)*a(1)

        def povrsina(a,b):
            return a * b

```

Pitanje 13. *

Prikazan je sljedeći kod.

```
a = formirajlistu(10)
print(a)
```

Odaberite među ponuđenim definicijama funkcija , funkciju koju bi ste dodali postojećem kodu koji generira i ispisuje listu parnih brojeva manjih od 10.

```
def formirajlistu(n):
    return(range(2,n,2))

def formirajlistu(n):
    l = list(range(2,n,2))
    return l

def formirajlistu(n):
    for i in range(2,n,2):
        lista.append(i)
        return lista

    def formirajlistu(n):
     lista = []
     for i in range(2,n,2):
         lista.append(i)
     return lista

    def formirajlistu(n):
       l = list(range(2,n,2))
```

Pitanje 14. **

Prikazan je sljedeći kod.

```
a = []
formirajlistu(a,10)
print(a)
```

Dovršiti program stvaranjem liste od 10 cijelih brojeva i ispišite listu . Koja od sljedećih definicija funkcije formirajlistu () će biti ispravna i dati odgovarajući rezultat?

```
def formirajlistu(lista,n):
 x = int(input())
 for i in range(n):
    lista.append(x)
 return lista

def formirajlistu(lista,n):
 for i in range(n):
  x = int(input())
          lista.append(x)
 return lista

def formirajlistu(lista,n):
 for i in range(n):
  x = int(input())
  lista[i] = x
 return lista

def formirajlistu(a,10):
 for i in range(10):
  x = int(input())
  a[i] = x
 return a
```

Pitanje 15.

Što je definirano sljedećim kodom?

```
def ispisiVeci(broj):
 print(broj + 1)
```

A. Postupak koji ispisuje broj koji je za jedan veći od zadanog broja.
B. Definicija podataka je nepotpuna. .
C. Niti jedan od ponuđenih odgovora nije točan.

Načini ispisa više varijabli istovremeno – Formatirani ispis

Povezivanje ispisa više podataka (varijabli) pomoću znaka " + "

Međutim, ovdje je jako važno napomenuti da svi podaci trebaju biti stringovi ili pretvoreni u string pomoću funkcije str(). Još jedna dobra stvar je što se ovim načinom može više podataka povezati i spojiti u jednu varijablu tipa string.

```
a=4
b=5
# Ispis pomoću zareza( , )
print("Zbroj brojeva", a , "i" , b , "je" , a+b)

# Ispis  pomoću ( + )
print(" Zbroj brojeva " + str(a) +  "i"  + str(b) +  " je "  +str(a+b))
```

U oba primjera ćemo dobiti Zbroj brojeva 4 i 5 je 9

No, kako se ne bi previše brinuli oko toga da li je sve pretvoreno u string, postoje još neki načini ispisa. Prikazat ćemo dva načina, puno fleksibilnija. Prvi način prikazan je na slici:

```
a=4
b=5
x="broj"
print("Zbroj brojeva %d i %d je % i." %(a, b , a+b))
# ispis  Zbroj brojeva 4 i 5 je 9
```

```
Print("Broj  % i  je  prirodan broj:" %a)
#ispis Broj 4 je prirodan broj
```

```
Print("Pozdrav od Pythona  %s"  %x)
#ispis Pozdrav od Pythona
```

Ovdje se unutar teksta (navodnika) postavljaju oznake; znak %, te nakon toga slovo koje nam govori o kojem se tipu radi. Nakon toga, unutar zagrada print() funkcije nalazi se nova zagrada ispred koje se opet stavlja znak %, dok se unutar te druge zagrade nalaze podaci koji se koriste pri ispisu.

Važno je napomenuti da poredak i tip mora odgovarati kako unutar rečenice, tako i unutar zagrada otkud se uzimaju podaci.

Slova predstavljaju tipove na sljedeći način:

i, d = integer (koristite jedan ili drugi ili oba)
f = float
s = string

Također, kao što je vidljivo na slici, kada se koristi samo jedan podatak, on se može nalaziti nakon znaka % bez zagrade, dok se kod više podataka obvezno mora staviti zagrada.

Naučit ćemo još jedan način ispisa za koji bi se moglo reći da je univerzalan i doista je pomoću njega moguće ispisati sve, kako god nam odgovara.

Takav ispis omogućava nam funkcija .format(). Ovo je jedna od jako važnih funkcija i pokušajte ju zapamtiti, jer ćete ju pri ispisu, pogotovo kod tablica i sličnih stvari, često koristiti.

Možda će vam ispisivanje pomoću .format() funkcije u početku izgledati komplicirano, ali kada jednom shvatite princip više neće biti problema. Primjer ove funkcije možete vidjeti na sljedećoj slici:

a=4 b=5
print(" Zbroj brojeva { } i { } je { }." .format(a, b , a+b)) **# ispis** Zbroj brojeva 4 i 5 je 9
print("Zbroj brojeva { 1 } i { 0 } je { 2 }." .format(a, b , a+b)) **# ispis** Zbroj brojeva 5 i 4 je 9
Zašto? *Brojevi se čitaju redoslijedom kako su napisani u programu. Prvi broj ima memorijsku oznaku 0 , a drugi memorijsku oznaku 1. U primjeru oznaku 2 ima memorijska lokacija u koju će se pohraniti rezultat matematičke operacije.*

Unutar rečenice se postavljaju vitičaste zagrade koje predstavljaju mjesto gdje će se prikazati određeni podatak.

Nakon rečenice stavlja se točka uz *format()*. Unutar ove zagrade nalaze se podaci koji će biti prikazani.

Važno je naglasiti da će izostankom brojeva unutar vitičastih zagrada podaci biti prikazivati po redu u kojem su napisani unutar zagrada funkcije *.format()*, dok ih pomoću brojeva u vitičastim zagradama možete pozivati po kojem god redoslijedu želite.

Još jedna važna stvar kod ove funkcije je dužina određenog podatka. Ova stvar je vrlo korisna ako želimo svakom podatku odrediti koliko će se znakova odvojiti za njegovo prikazivanje.

U tom slučaju, nakon broja unutar vitičaste zagrade stavljamo dvotočku i broj koji će označavati koliko će znakova zauzimati prikazani podatak. Slobodno eksperimentirajte s veličinom, vidjet ćete razlike pri prikazivanju pri raznim veličinama, jer ulogu igra i veličina ostalih podataka unutar funkcije print().

Ponovimo vrste formatiranog ispisa:

a=4 b=5 print("Zbroj brojeva",a,"i",b,"je",a+b)	**Ovaj ispis smo već obradili u naredbi print**
a=4 b=5 print("Zbroj brojeva {} i{} je {}.".format(a,b,a+b))	**Korištenje vitičastih zagrada, svaka vitičasta zagrada predstavlja jednu varijablu u našem primjeru varijable (a i b), treća vitičasta zagrada je za prikaz rezultata. Koristimo naredbu .format(a,b,a+b)**
a=4 b=5 print("Zbroj brojeva {1} i{0} je {2}.".format(a,b,a+b))	**Brojevi u vitičastim zagradama predstavljaju redoslijed izvršavanja varijabli . Mi imamo varijablu a i b , želimo zbrojiti vrijednosti varijable b i a . Koristimo brojeve u vitičastim zagradama. Prva varijabla u zagradama je označena sa 0 to je varijabla a, druga varijabla označena je brojem 1 to je varijabla b, Rezultat je predstavljen trećom varijablom i ima broj 2.**
a=4 c=5 print("{:5}{:10}".format(a,b,a+b))	**Ovakav oblik predstavlja formatirani ispis gdje će varijabla a biti udaljena 5 mjesta od lijeve strane ekrana, a varijabla b 10 mjesta od lijeve strane.**
a=4 b=5 c=a+b print("Zaokruži na dvije decimale{:5.2f}".format(c))	**{0:5.2f}".format(c) – 2f – označava 2 decimalna mjesta {0:5} Znači da smo za prikaz decimalnog mjesta osigurali pet(5) mjesta u to ubrajamo i decimalnu točku. Uzmimo broj 10/3 =3.33333333 03.33 - {0:5.2f} rezervirali smo pet mjesta od kojih su dva decimalna , nula s lijeve strane broja nema nikakvo značenje. To smo učili na matematici.**

Primjer:

Pogledajte dati primjer i vidjet ćete pravo značenje formatiranih ispisa. Uz pomoć učitelja prokomentirajte svaki ispis.
 	
Poigrajte se malo sa različitim brojevimai vrstama ispisa
	**a=4
	b=5
	c=float(3.4566789)
	#ispis pomoću zareza (,)
	print("Zbroj brojeva",a,"i",b,"je",a+b)
	print("Zbroj brojeva {} i{} je {}.".format(a,b,a+b))
	print("Zbroj brojeva {1} i{0} je {2}.".format(a,b,a+b))
	print("Zaokruživanje na dvije decimale{:5.2f}".format(c))
	print("{:5}{:10}".format(a,b,a+b))**

Formatirani ispisi su vrlo korisni i koristit ćete ih kada ćete svoje rezultate obrade htjeti urediti tako da lijepo izgledaju na ekranu.

Primjer:

Želimo napraviti sljedeći formatirani ispis: **IME PREZIME** **------- -------------** **Ana Malić**	**print("{:5}{:10}".format("Ime","PREZIME"))** **print("{:5}{:10}".format("----","-------"))** **print("")** **print("{:5}{:10}".format("Ana","Malić"))**

Napravite samostalno neki formatirani ispis: npr. Neki jednostavni račun.

Podatkovne zbirke: nTorke, skupovi, rječnici

1. Zbirke sa slijednim smještajem elemenata**:**
   a) Znakovni niz ili stringovi
   b) Liste
   c) N-torke
2. Zbirke s raspršenim smještajem
   a) Skupovi
   b) Rječnici

N-torke:

N-torka može sadržavati više elemenata različitog tipa podataka

Često se koriste za prenošenje(vraćanje) više vrijednosti iz funkcija

nTorka=("Skola",2019) – elementi u ntorki su omeđeni okruglim zagradama, a odvojeni zarezom.

Elementima nTorke pristupa se pomoću indeks. (kao u listama)

SKUPOVI:

a) Skupovi odgovaraju matematičkom pojmu skupa i nad njima se primjenjuju iste zakonitosti kao i matematici
b) Sastoje se od strogo različitih vrijednosti
c) Skupovi onemogućuju ponavljanje elemenata

Npr.

string="škola" skup=set(string) print(skup) {"k","š","a","l","o"} len(string) rezultat= 5 len(skup) rezultat=5	a) Funkcija set() pretvara znakovni niz(string) ili listu u skup b) Elementi u skupu su omeđeni vitičastim zagradama c) Redoslijed elemenata u skupu je proizvoljan i elementima se ne može pristupiti pomoću indeksa

1. Nad skupovima su definirane različite standardne matematičke operacije, kao što su unija, presjek, razlika,....

skup1={1,2,3,4,5} skup2={3,4,5,6,7}	
print(skup1 I skup2) - unija	{1,2,3,4,5,6,7}
print(skup1 & skup2}- presjek	{3,4,5}
print(skup1 – skup2) - razlika	{1,2}

Često skupovi nastaju iz lista. Time možemo jednostavno eliminirati ponavljajuće elemente.

```
ocjene=[1,1,1,2,3,3,4,4,4,5,5,5,5,5]
skup=set(ocjene)
print(skup)
{1, 2, 3, 4, 5}
```

Primjer_1:

Poznata su nam imena svih igrača koji su postizali golove tijekom jedne prijateljske utakmice redoslijed kako su postigli golove, lako možemo odrediti skup svih igrača koji su ušli u listu strijelaca. Naime, ako formiramo skup na temelju imena strijelaca, duplikati će se automatski izbrisati.

```
strijelci = {"Mesi", "Ronaldo", "Mesi", "Ibrahimović", "Ibrahimović", "Nejmar", "Nejmar"}
print(strijelci)
golovi = ["Mesi", "Ronaldo", "Mesi", "Ibrahimović", "Ibrahimović", "Nejmar", "Nejmar"]
strijelci = set(golovi)
print(strijelci)
```

Primjer_2:

Djevojke koje treniraju dva sporta

Jedan skup sadrži djevojke koje treniraju ritmičku gimnastiku, a drugi djevojke koje treniraju odbojku. Identificirajte skup djevojaka koje treniraju oba sporta, skup djevojaka koje treniraju barem jednu od njih i skup djevojaka koje treniraju samo odbojku.

```
ritmicka = {"Ana", "Marica", "Ivana", "Gordana"}
odbojka = {"Tara", "Nada", "Marica", "Ivana", "Aleksandra"}
dva_sporta = ritmicka & odbojka
bar_jedan_sport = ritmicka | odbojka
samo_odbojka = odbojka - ritmicka
print(dva_sporta)
print(bar_jedan_sport)
print(samo_odbojka)
```

Rječnici

a) Rječnici sadrže parove vrijednosti : ključ – pripadna vrijednost.

b) Vrijednosti se pohranjuju i dohvaćaju na temelju ključa, a ne indexa

c) Nazivi ključa su znakovni nizovi

rjecnik={“naziv“: “Škola“,“prog“:“Python“,“god“:2019“}

print(rjecnik)	{“naziv“: “Škola“,“prog“:“Python“,“god“:2019“}
print(rjecnik.keys()) – dohvaćanje svih ključeva rjecnika	{“naziv“,“prog“,“god“}
print(rjecnik.values()) – dohvaćanje svih vrijednosti rječnika	{“Škola“,“Python“,“2019“}
print(len(rjecnik))	3
Print(rjecnik[“naziv“])	Škola

Petlje u rječnicima

rjecnik={"mislav":"dobar","marko":"loš","filip":"zao"} for kljuc,vrijednost in rjecnik.items(): print(vrijednost+" "+kljuc)	items – predstavlja i ključeve i vrijednosti

```
vocarna={1001:"Jabuka",1002:"Kruška",1003:"jagoda",1004:"Banana",1005:"kivi"}
for kljuc,vrijednost in vocarna.items():
   print(kljuc," - ",vrijednost)
```

Uvjeti u rječnicima

```
rjecnik={1001:"Ana",2019:"Ivan"}

kljuc=int(input("Unesite ključ: "))

if kljuc in rjecnik:

   print("Ključ je pronađen")

   print("Ime:",rjecnik[kljuc])

if not kljuc in rjecnik:

   print("Ključ nije pronađen")
```

Primjer_1:

Automobili u oglasniku imaju prikazane cijene. Mi želimo odrediti cijenu automobila na temelju njegovog modela. Napišite program koji na temelju modela automobila koji unosite , određuje njegovu cijenu (cijeli broj).

Ključan dio za učinkovito rješavanje ovog zadatka je predstavljanje kataloga cijena automobila pomoću rječnika.

```
cijene_auta = {"fiat 500l": 11990, "renault clio": 9650, "toyota corolla": 13990}
auto = input("Unesi model automobila:")
print(cijene_auta[auto])
```

Primjer_2:

Poznate su geografske koordinate nekoliko velikih europskih gradova. Za određeni naziv grada odredite njegove geografske koordinate. Navedite posebno zemljopisnu dužinu i širinu.

```
gradovi = {"Zagreb": (44.7866, 20.4489),
        "Budimpešta": (47.4979, 19.0402),
        "Beč": (48.2082, 16.3738),
        "Bratislava": (48.1486, 17.1077)}
grad = input("Unesi ime grada: ")
# ispravi naredni red tako da se iz rječnika pročitaju koordinate grada
koordinate = ()
print(koordinate)
# dopuni naredna dva reda tako da se ispišu geografska širina i dužina
print("Geografska širina: ")
print("Geografska dužina: ")
```

Primjer_3:

Nadopunite rječnik koji svakoj zemlji dodjeljuje popis koji sadrži tri najveća grada. Upotrijebite ga u programu koji ispisuje gradove na temelju naziva zemlje

```
najveci_gradovi = {}

zemlja = input("Unesi naziv zemlje:")

if zemlja in najveci_gradovi:

   print(najveci_gradovi[zemlja])

else:

   print("Nepoznata zemlja:", zemlja)
```

Primjer_4:

Definirajte rječnik koji imenu svakog mjeseca dodjeljuje njegov broj dana tijekom 2019. godine.
Za učitani naziv mjeseca, ispišite njegov broj dana.
Nadopunite programski kod.

```
broj_dana = {"siječanj": 31, "veljača": 28}  # dopuni ovaj red

mjesec = int(input("Unesi naziv mjeseca"))

if mjesec in broj_dana:

   print()  # dopuni ovaj red

else:

   print("Nepoznati mjesec", mjesec)
```

Zadaci za vježbu:

Pitanje 1.

Na koji način Python definira skup koji sadrži brojeve -3, 14, 7, 22?

Odaberite točan odgovor:

A. s = [-3, 14, 7, 22]

B. s = (-3, 14, 7, 22)

C. s = {-3, 14, 7, 22}

Pitanje 2. *

Što će biti rezultat izvršavanja sljedećeg koda?

```
s = {-1, 57, 20, -1}
print(s)
```

Pitanje 3. *

Što će biti rezultat izvršavanja sljedećeg koda?

```
s = [-5, 43, 18, -5]
print( set(s) )
```

Pitanje 4.

Što će Python ispisati nakon izvršenja sljedećeg koda?

```
A = {34, 38, 16, 25, 30}
B = {33, 41, 10, 45, 21, 27}
C = A & B
print( C )
```

Pitanje 5.

Što će Python ispisati nakon izvršenja sljedećeg koda?

```
A = {33, 39, 15, 26, 28}
B = {32, 42, 44, 46}
C = A | B
print( C )
```

Pitanje 6.

Što će Python ispisati nakon izvršenja sljedećeg koda?

```
a = {49, 21, 24, 28, 31}
print( len(a) )
```

Pitanje 7.

Što će Python ispisati nakon izvršenja sljedećeg koda?

```
a= {32, 1, 41, 12, 46, 19, 27, 30}
b = {33, 1, 41, 43, 48, 22, 28, 29}
c = A & B
d = C - {1}
print( len(d) )
```

Pitanje 8.

Što će Python okolina ispisati nakon izvršenja sljedećeg koda?

```
p = (177,75)
print(p[1])
```

Pitanje 9.

Što će biti rezultat izvršavanja zadanog programa?

```
s = {'a8': 12.0}
lista = []
for x in s:
  lista.append(s[x])
print(lista)
```

PETLJA U PETLJI – DVOSTRUKA PETLJA

Petlje se mogu izvršavati jedna unutar druge. Prva petlja određuje koliko će se puta odvijati druga petlja.

Primjena petlje u petlji u programskom modulu

Napravimo program koji će u četiri reda ispisati po pet zvjezdica, jedno ispod drugog s jednim redom praznine između njih.
 	
Crveno je označen blok brojača **i**
Zeleno je označen blok brojača j
 	
Pojašnjenje:
 	
Prva petlja ispisuje stupac, a druga petlja ispisuje po pet zvjezdica u svakom od četiri retka. Između svakog retka je prazan red.
	`for i in range(4):`
	`    for j in range(5):`
	`        print(end="*")`
	` print(" ")`

Sljedeći programski kod dat će nam sljedeći ispis:
 	
?
??
???
 	
Izmjenite naredbu print i upišite
print(end=" ? ")
	`for i in range(4):`
	`    for j in range(i):`
	`        print(end=" ")`
	`    print(" ")`

Koraci u petlji :

	i	j	ISPIS
1. korak	1	1	**1**
2. korak	2	**1**	**1 2**
3. korak	2	**2**	
gotova petlja po *j* prijelaz u novi red			
4. korak	3	**1**	**1 2 3**
5. korak	3	**2**	
6. korak	3	**3**	
gotova petlja po *j* prijelaz u novi red			
itd			

Primjer 1:

Napravimo program za izradu tablice množenja do 10.
Pomoću formatiranog ispisa, uredite ispis tako da tablica množenja lijepo izgleda.
	<pre>for i in range(1,11):
	   for j in range(1,11):
	      print(i*j,end=" ")
	   print(" ")</pre>

ALGORITMI SORTIRANJA

Jednostavan primjer sortiranja – buuble sort

```
# sortiranje niza
niz = [2, 7, 1, 0, 6]
for i in niz:   # na ovaj način for petlja prođe kroz čitav niz
   print(i)   # varijabla i iz for postaje u svakom ciklusu drugi element

n = len (niz)
for i in range (0, n - 1):
   for j in range (i + 1, n):
      if (niz[i] > niz[j]):  #  usporedi sa prvi iz podniza (index i) sa svima iz podniza
         b = niz[i]     #  zamjena mjesta
         niz[i] = niz[j]
         niz[j] = b

print("\n Ispis članova niza prije sorta")
for k in niz:   # na ovaj način for petlja prođe kroz čitav niz
   print(k)    # k varijabla k iz for postaje u svakom ciklusu drugi element
```

Plavo označen dio koda najčešće ćete koristiti pri izradi programa pa ga dobro proučite.

Buuble sort: primjer rada algoritma.

1. Počinjemo sa prvim elementom liste (indeks=0), uspoređuje se prvi element liste sa sljedećim elementom liste (indeks=1)
2. Ako je element (indeks=0) veći od sljedećeg elementa liste (indeks=1), zamjeni ih
3. Ako je element (indeks=0) manji od elementa sa (indeks=1), pomakni se na sljedeći element (indeks=2). Ako je zadovoljen ovaj uvjet , vrati se na prvi korak(1.)

Primer liste: [10,15,4,23,0]

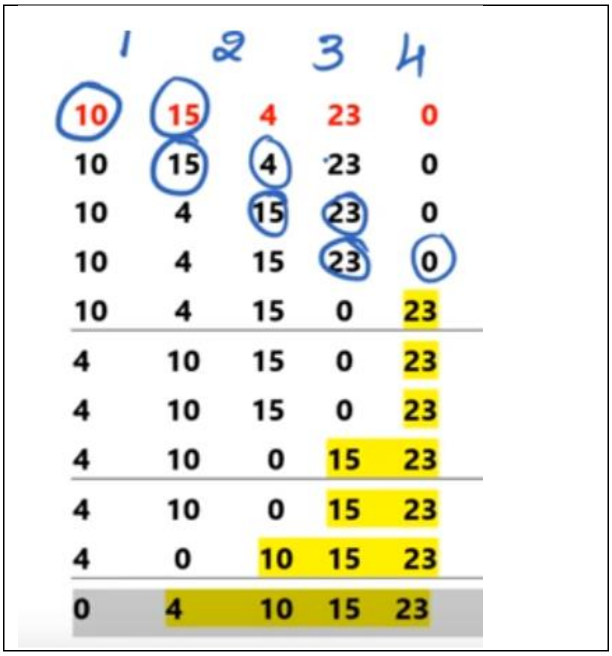
Prvi korak

10	15	4	23	0
10	15	4	23	0
10	4	15	23	0
10	4	15	23	0
10	4	15	0	23
4	10	15	0	23
4	10	15	0	23
4	10	0	15	23
4	10	0	15	23
4	0	10	15	23
0	4	10	15	23

Drugi korak

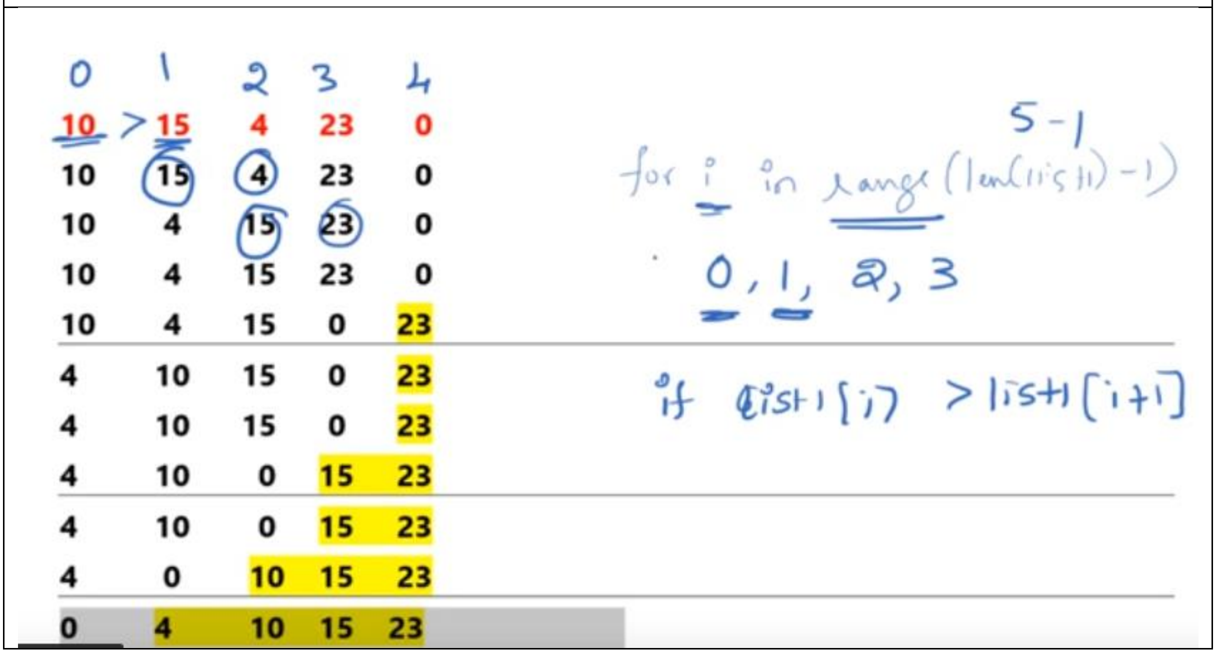
10	15	4	23	0
10	15 >	4	23	0
10	4	15	23	0
10	4	15	23	0
10	4	15	0	23
4	10	15	0	23
4	10	15	0	23
4	10	0	15	23
4	10	0	15	23
4	0	10	15	23
0	4	10	15	23

Treći korak



	1	2	3	4
10	15	4	23	0
10	15	4	23	0
10	4	15	23	0
10	4	15	23	0
10	4	15	0	23
4	10	15	0	23
4	10	15	0	23
4	10	0	15	23
4	10	0	15	23
4	0	10	15	23
0	4	10	15	23

Četvrti korak



0	1	2	3	4
10 >	15	4	23	0
10	15	4	23	0
10	4	15	23	0
10	4	15	23	0
10	4	15	0	23
4	10	15	0	23
4	10	15	0	23
4	10	0	15	23
4	10	0	15	23
4	0	10	15	23
0	4	10	15	23

Ovo je primjer samo jedne usporedbe (indeks=0 i indeks=1), međutim mi želimo sortirati sve elemente liste , za to moramo koristiti dvostruku petlju.

Primjer samo jedne zamjene:

list1=[10,15,4,23,0] print(“nesortirana lista: “,list1)	**Stvaranje liste**

for i in range(len(list1)-1)
        if list1[i]>list1[i+1]:
            list1[i],list1[i+1]=list1[i+1],list1[i]
	Koraci sortiranja

print(“Sortirana lista: “,list1)	**Konačni rezultat**

```
list1=[10,15,4,23,0]
print(“nesortirana lista: “,list1)

for j in range(len(list1)-1):
    for i in range(len(list1)-1):
        if list1[i]>list1[i+1]:
            list1[i],list1[i+1]=list1[i+1],list1[i]
print(“Sortirana lista: “,list1)
```

```
list1=[10,15,4,23,0]
print(“nesortirana lista: “,list1)

for j in range(len(list1)-1):
    for i in range(len(list1)-1):
        if list1[i]>list1[i+1]:
            list1[i],list1[i+1]=list1[i+1],list1[i]
            print(list1)
print(“Sortirana lista: “,list1)
```

```
list1=[10,15,4,23,0]
print(“nesortirana lista: “,list1)

for j in range(len(list1)-1):
    for i in range(len(list1)-1):
        if list1[i]>list1[i+1]:
            list1[i],list1[i+1]=list1[i+1],list1[i]
            print(list1)
        else:
```

`print(list1)`
`print()`
`print("Sortirana lista: ",list1)`	

Primjer unosa podataka u listu i sortiranje liste

```
list1=[]
num=int(input("Koliko elementa želite unijeti   "))
print("Unesite elemente u listu  ")
for k in range(num):
    list1.append(int(input()))

print("nesortirana lista: ",list1)

for j in range(len(list1)-1):
    for i in range(len(list1)-1):
        if list1[i]>list1[i+1]:
            list1[i],list1[i+1]=list1[i+1],list1[i]
            print(list1)
        else:
            print(list1)
    print()
print("Sortirana lista: ",list1)
```

Buuble sort primjenom funkcije	**list1[i],list1[i+1]=list1[i+1],list[i]** ***isto je kao i :*** **temp=list[j]** **list[j]=list[j+1]** **list[j+1]=temo**
def buuble_sort(list): **for i in range(len(list)-1,0,-1):** **for j in range(i):** **if list[j]>list[j+1]:** **temp=list[j]** **list[j]=list[j+1]** **list[j+1]=temp** **list=[5,3,8,6,7,2]** **buuble_sort(list)** **print(list)**	**Pojašnjenje:** **Primjer: imamo dvije čaše sa dva različita soka (čaša1=kola, čaša2=fanta).** **U čašu u kojoj je kola želimo staviti fantu. Kako bi ste to izveli. Naravno uzeli bi ste treću čašu koja bi vam poslužila za privremenu radnju zamjene sokova.**

INSERTION SORT

Koraci algoritma:

1.Svaki element u listi se uspoređuje s onim prije sebe, ako je manji elementi zamjenjuju mjesta.

Primjer:

	**Uspoređujemo element s indeks=1 , da li je veći od indeks=0, ako je prelazimo na element s indeks=2**
	**Kao što vidimo indeks=2 je manji od indeks=1 i zamjenjuju mjesta, zatim se uspoređuje da li je indeks=1, manji od indeks=0, ako je zamjene mjesta**
	**Provjeravamo indeks=3 da li je manji od indeks=2,indeks=1 i indeks=0, ako je zamjenjuju mjesta**
	**Provjeravamo da li je indeks=4 manji od indeks=3, ako nije nalazi se na dobrom mjestu**
	**Provjeravamo indeks=5, da li je manji od indeks=4 , ako je zamjenjuju mjesta.** **I tako sve do indeks=0.**

```
# Primjer za Insertion Sort

# Definiramo funkciju inseertinSort
def insertionSort(arr):
        # Kretanje kroz listu
        for i in range(1, len(lista)):

                key = lista[i] # Uzimamo element s indeksom=1

                # Pomiče elemente liste, lista[0..i-1], koji su
                # veći od key, na prvu poziciju( indeks=0)
                j = i-1
                while j >=0 and key < arr[j] :
                                lista[j+1] = lista[j]
                                j=j- 1
                lista[j+1] = key

# primjer liste
lista = [10, 9, 2, 5, 6]
insertionSort(lista)
print ("Sortirani niz:")
for i in range(len(lista)):
        print ("%d" %lista[i])
```

Primjer_2: Isti primjer kao gore samo s malo manje koda.

```
def insertion_sort(lista):
   for index in range(1,len(lista)):
      trenutni_element=lista[index]
      pos=index
      while trenutni_element<lista[pos-1] and pos>0:
         lista[pos]=lista[pos-1]
         pos=pos-1
      lista[pos]=trenutni_element


lista=[4,1,5,2,9,8]
insertion_sort(lista)
print(lista)
```

Datoteke

Što su to datoteke i čemu služe??

Datoteke nam služe za pohranjivanje raznih sadržaja koji se u računalu trebaju trajno čuvati.

Datoteke može stvoriti i popuniti jedan program i nakon toga njezin sadržaj može dohvatiti neki drugi program.

Datoteke u računalnim sustavima imaju dvojaku ulogu:

a) One služe za trajno pohranjivanje svih oblika informacija
b) Pomoću njih se može obavljati razmjena informacija između programa.

Datoteke imaju nastavak **.txt** . Takve datoteke možemo pripremiti u bilo kojem tekstualnom editoru.

Najbolje je koristiti NOTEPAD ili BLOK EDIITOR, može se napraviti i u Wordu , ali pazite da nastavak uvijek bute **TXT.**

Za početak kreirajte jednu mapu i u pohranjujte svoje datoteke i izvorne kodove programa.

Za početak, ako želimo nešto spremiti u datoteku, prvo ju trebamo otvoriti.

Za to koristimo funkciju *open()*. Ova funkcija prima kao prvi argument string, tj. ime datoteke, dok je drugi argument način otvaranja datoteke. Oboje se svakako mora pisati unutar navodnih znakova. Primjer je prikazan ispod:

datoteka = open("datoteka.txt", "w")

Jedino što eventualno još nije jasno je način otvaranja datoteke odnosno mod. U ovom slučaju je to *w* koje označava da se radi o pisanju u datoteku. Osim tog moda, postoji ih još nekoliko; koristit ćemo w (engl. write) za pisanje, r (engl. read) za čitanje, wr (engl. write & read)

Ako želimo nešto zapisati u datoteku, za to koristimo metodu *.write()*. U nastavku je prikazan primjer:

datoteka.write("Ovo je prvi redak :)")

I na kraju, kako bi se sve što smo dosad napisali spremilo, potrebno je datoteku zatvoriti. Za to koristimo metodu *.close()*.

datoteka.close()

Ono što treba napomenuti je da se datoteka sprema u onu mapu u kojem se nalazi i sam program.

Primjer_1.

Kreiranje tekstualne datoteke

```
file=open(„datoteka.txt“,“w“)

file.write("početak rada“)

file.write("nova linija teksta“)

file.write("druga linija teksta“)

file.close()
```

Čitanje tekstualne datoteke

Nekoliko načina čitanja

1. Čitanje cijele datoteke **print file.read()**
2. Čitanje do određene pozicije **print file.read(5)**
3. Čitanje određenog retka **print file.readline(3)**
4. Čitanje svih redaka jedan za drugim (u jednoj liniji , kao lista) **print file.readlines(),**

Razlika READ,REDLINE, READLINES

Primjer: sadržaj tekstualne datoteke

Prva linija teksta
Druga linija teksta
Treća linija teksta
Četvrta linija teksta

Za čitanje datoteka koristimo dodatak "r"

READ	ispisat će sve četiri linije teksta, read čita znakove pa ako stavimo read(5) , ispisat će prvih pet znakova "PRVA_"
READLINE	Ispisat će samo prvu liniju teksta
READLINES	Ispisat će se sve linije teksta, ali u jednom retku, svaki red će imati oznaku \n za oznaku odlomka. Cijeli tekst će pretvoriti u listu ["Prva linija teksta\n", "Druga linija teksta\n", "Treća linija teksta\n, "Četvrta linija teksta\n"]

Za upis podataka u datoteku koristimo dodatak "w ili a"

w- ukoliko u datoteci imamo podataka novi podaci će obrisati stare

a- Ukoliko u datoteci imamo podataka novi podaci će biti dodani u datoteku bez brisanja postojećih.

Svaki podatak koji unesete u datoteku imat će tip podataka STRING.

WRITE	Koristi se za unos jednog retka podataka
WRITLINES	Koristi listu , za unos podataka ["Prva linija teksta\n", "Druga linija teksta\n", "Treća linija teksta\n, "Četvrta linija teksta\n"]

Također, ako ne želite provjeravati uneseni tekst pomoću tekst editora, to možete i pomoću Pythona. Prije smo spomenuli način rada za pisanje, ali postoji način rada i za čitanje. Stoga, ako pri otvaranju datoteke kao drugi argument stavimo *r*, moći ćemo pročitati sadržaj datoteke (datoteka mora prethodno biti zatvorena i tek se nakon toga može otvoriti u novom načinu rada).

datoteka = open(“datoteka.txt”, “r”)

Sada kada smo datoteku otvorili, potrebno je pročitati njen sadržaj. To se radi pomoću metode *.read()*. Kako bismo to prikazali na ekranu, imamo više načina; prvi je da se sve stavi u funkciju *print()*:

print(datoteka.read())

ili možemo sadržaj spremiti u novu varijablu koju onda kasnije koristimo kroz program:

sadrzaj = datoteka.read()
print(sadrzaj)

VAŽNO: Ako želite pisati više teksta u datoteku, a to ne radite odjednom već u više navrata, *NEMOJTE* koristiti način rada *w*. Razlog tome je što ovaj način rada briše stari sadržaj i piše novi. U tom slučaju koristite način rada *a* (engl. append) koji na kraj datoteke dodaje nove vrijednosti odnosno tekst.

Kod čitanja i pisanja datoteka također trebamo paziti na jednu stvar, a radi se o novom retku. Ako se tekst proteže kroz nekoliko redaka i razlog tomu je korištenje tipke Enter za novi redak, na tom mjestu se generira znak “\n”. Kako kod manipuliranja sadržajem ne bismo imali problema, onda se zna koristiti metoda *.rstrip()* koji s desne strane retka ukloni znak “\n”.

Primjer_2:

Primjer unosa i ispisa datoteke pomoću Pythona

dat=open("edita.txt","w") dat.write("Početak rada\n") dat.write("Drugi redak\n") dat.write("Nova linija programa\n") dat.close()	Unosa podataka u datoteku. Primjena naredbe write
x=open("edita.txt","r") z=x.read() print(z)	Ispis sadržaja datoteke edita.txt Načini ispisa iz datoteke : - Čitanje cijele datoteke read() - Čitanje do određene pozicije read(5) - Čitanje određenog retka readline(3) - Čitanje svih redaka jedan za drugim readlines()

Zadatak za vježbu:

Napravite tekstualnu datoteku **podaci.txt** te upišite svoje ime i prezime u tu datoteku, te nakon toga pročitajte sadržaj datoteke i spišite u pythonu, u jednom retku vaše ime, a u retku ispod prezime.

Prije nego krenemo sa zadacima , morate razumjeti da podaci koje se nalaze u nekoj datoteci pohranjeni su kao string. , nad tim podacima ne možete izvršavati nikakve matematičke operacije bez obzira ako su u datoteci uneseni brojevi. Program ih čita kao znakove.

U tu svrhu koristimo metodu SPLIT i pretvaranje stringova(znakova) u broj.

Primjer:

tekst="Učimo datoteke u Pythonu"
tekst.split()

rezultat:
["Učimo", "datoteke", "u", "Pythonu"]

Kao što možemo vidjeti metoda SPLIT() često se koristi kada se više podataka unosi u istom retku (s jednom input funkcijom) . Podaci koji su uneseni imaju kao tip podataka string .

Tj. tekst koji upišemo u jednom retku ona pretvara u listu. Rad s listama smo učili i malo se podsjetimo.

Kao što znamo nad stringovima (ili znakovima) ne možemo izvršavati matematičke operacije. Unosimo li niz brojeva kao npr: 2,5,3,6 .U naredbi split(,) možemo koristiti i znakove odvajanja.

unos=input("Unesi četiri broja odvojena zarezom")
b1,b2,b3,b4=unos.split(",")
print(b1+b2+b3+b4)
rezultat je: 2536 spojili smo stringove i ništa više.

Želimo li s tim brojevima izvršavati razne matematičke operacije , string moramo pretvoriti pretvoriti u tip (int ili float) :
b1=int(b1)
b2=int(b2)
b3=int(b3)
b4=int(b4)

sada:
print(b1+b2+b3+b4) dobit ćemo rezultat 16.

Pretvaranje teksta u listu , moramo koristiti metodu split().
Pretvaranje stringa iz liste u broj moramo koristiti (int ili float)

Pri ispisu podataka iz datoteke, obavezno ćete koristiti formatirani ispis, najčešće vitičaste zagrade { }. Ponovite formatirane ispise !

Zadaci s rješenjima :

Primjer_1:

Pomoću nekog tekst editora u datoteku Linda.txt upisati brojeve koji slijede.

Napišite program koji će u izlaznu datoteku Linda2.txt ispisati samo parne brojeve

Podaci u datoteci Linda.txt	Sadržaj izlazne datoteke Linda2.txt
1 5 12 16 22 11	12 16 22

```
ulaz=open("linda.txt","r")
izlaz=open("linda2.txt","w")
redci=ulaz.readlines()
for s in redci:   # Pretražujemo sadržaj ulazne datoteke
   b=s.split()    # svaku riječ iz ulazne datoteke pretvara u listu i string
   for i in range(len(b)): # čitamo svaki element iz liste
      b[i]=int(b[i])  # Elemente iz liste pretvaramo u cijele brojeve
      if (b[i]%2==0):
         izlaz.write("{}  ".format(b[i])) #važnost formatiranog ispisa
ulaz.close()
izlaz.close()

#Ispis iz datoteke
izlaz=open("linda2.txt","r")
red=izlaz.read()
for i in red:
   print(i,end="")
```

Primjer_2:

U datoteku erik.txt unesite pomoću nekog tekst editora podatke kako slijedi.

U izlaznu datoteku erik1.txt, u prvom retku ispisat će se najveći broj, u drugi redak najmanji broj, u treći redak ispisati zbroj svih brojeva.

Sadržaj ulazne datoteke erik.txt	Sadržaj izlazne datoteke erik1.txt
14 -3 8 10 -12	14 -12 -3

```
ulaz=open("erik.txt","r")
izlaz=open("erik1.txt","w")
redci=ulaz.readlines()
for s in redci:
    b=s.split()
    for i in range(len(b)):
        b[i]=int(b[i])
    izlaz.write("{}\n {}\n{}\n ".format(max(b),min(b),sum(b)))
ulaz.close()
izlaz.close()

#Ispis iz datoteke
kk=open("erik1.txt","r")
red=kk.read()
for i in red:
    print(i,end="")
```

Primjer_3.

U datoteku erik.txt unesite pomoću nekog tekst editora podatke kako slijedi.

Napišite program koji će u izlaznoj datoteci Marko.txt napisati zbroj svakog retka posebno

-15 16 -18 2 -5 4 -11 23 3 -23 12 -13 2 1 -3 4 5 13	-15 14 -21 19

```
ulaz=open("Erik.txt","r")
izlaz=open("Marko.txt","w")
redci=ulaz.readlines()
for s in redci:
    b=s.split()
    for i in range(len(b)):
        b[i]=int(b[i])
    izlaz.write("{}\n ".format(sum(b)))
ulaz.close()
izlaz.close()

#Ispis iz datoteke
kk=open("Marko.txt","r")
#red=kk

for i in kk:
    print(i,end="")
```

Zadaci za vježbu:

Zadatak_1:

U datoteku brojevi.txt spremi sljedeće brojeve:

6 18 4 19 15 11 8 23 17 12 163 6 17 18 22 1 4	- U izlaznu datoteku rezultat.txt ispišite: a) Najveći i najmanji broj za svaki redak b) Ispisati sve parne brojeve u svakom retku c) Ispisati neparne brojeve u svakom retku d) Ispisati u svakom retku brojeve djeljive sa 2 e) Izračunajte sumu zbroja brojeva u svakom retku.

Analiza zadataka:

Zadatak_1:

Sa stranice preradovic.hr – 8.razred, otvorite mapu datoteke:

Otvorite datoteku primjer_unos_ispis_1.py , analizirajte program

Zadatak_2:

Otvorite datoteku primjer_unos_ispis_2.py ,u ovom primjeru je rješenje sljedećeg zadatka:

U datoteku Guma.txt spremi podatke za n učenika: njihov uspjeh, ocjenu iz informatike i njihovu masu(težinu učenika). Te podatke pročitaj iz datoteke i ispiši:

a) Najbolji uspjeh
b) Mase sortirano od manje prema većoj
c) Umnožak ocjena iz informatike
d) Koliko učenika ima uspjeh 4, ocjenu iz informatike 5 i masu jednaku 70 kg
e) Umnožak parnih masa u intervalu <60,70]

Dobivene rezultate smjesti u tekstualnu datoteku Guma3.txt