**PANEVROPSKI UNIVERZITET APEIRON**

**FAKULTET INFORMACIONIH TEHNOLOGIJA**

**Vanredne studije**

**Smjer „Programiranje i softversko inženjerstvo“**

**Predmet:**

**SKRIPTNI JEZICI I PROGRAMIRANJE**

**15 zadataka u programskom jeziku “Python”**

**(seminarski rad)**

**Predmetni nastavnik**

**Doc.dr Tijana Talić**

**Student**

**Mirza Abdijanović**

Indeks br. 93-22/FITIIT1-pro-180

Banja Luka, Jun 2025

Sadržaj

[ZADATAK 1: Dijeljenje dva broja 4](#_Toc201530336)

[Kod zadatka 1: 4](#_Toc201530337)

[Primjer izvršavanja koda zadatka 1: 5](#_Toc201530338)

[ZADATAK 2: Provjera parnog broja 5](#_Toc201530339)

[Kod zadatka 2: 5](#_Toc201530340)

[Primjer izvršavanja koda zadatka 2: 6](#_Toc201530341)

[ZADATAK 3: Brojanje pojavljivanja svakog karaktera u stringu 6](#_Toc201530342)

[Kod zadatka 3: 6](#_Toc201530343)

[Primjer izvršavanja koda zadatka 3: 7](#_Toc201530344)

[ZADATAK 4: Prebrojavanje praznih mjesta, brojeva, slova i specijalnih karaktera u unesenom stringu 7](#_Toc201530345)

[Kod zadatka 4: 8](#_Toc201530346)

[Primjer izvršavanja koda zadatka 4: 8](#_Toc201530347)

[ZADATAK 5: Izračunavanje faktorijela broja korištenjem while petlje 8](#_Toc201530348)

[Kod zadatka 5: 9](#_Toc201530349)

[Primjer izvršavanja koda zadatka 5: 9](#_Toc201530350)

[ZADATAK 6: Brojanje samoglasnika u unesenoj rečenici 9](#_Toc201530351)

[Kod zadatka 6: 10](#_Toc201530352)

[Primjer izvršavanja koda zadatka 6: 10](#_Toc201530353)

[ZADATAK 7: Provjera da li je unesena riječ palindrom 10](#_Toc201530354)

[Kod zadatka 7: 11](#_Toc201530355)

[Primjer izvršavanja koda zadatka 7: 11](#_Toc201530356)

[ZADATAK 8: Izračunavanje prosječne ocjene studenta 11](#_Toc201530357)

[Kod zadatka 8: 12](#_Toc201530358)

[Primjer izvršavanja koda zadatka 8: 12](#_Toc201530359)

[ZADATAK 9: Generisanje Fibonačijevog niza 13](#_Toc201530360)

[Kod zadatka 9: 13](#_Toc201530361)

[Primjer izvršavanja koda zadatka 9: 13](#_Toc201530362)

[ZADATAK 10: Kalkulator 14](#_Toc201530363)

[Kod zadatka 10: 14](#_Toc201530364)

[Primjer izvršavanja koda zadatka 10: 15](#_Toc201530365)

[ZADATAK 11: Pogađanja broja 15](#_Toc201530366)

[Kod zadatka 11: 15](#_Toc201530367)

[Primjer izvršavanja koda zadatka 11: 16](#_Toc201530368)

[ZADATAK 12: Rad sa fajlom - zapis i čitanje 16](#_Toc201530369)

[Kod zadatka 12: 17](#_Toc201530370)

[Primjer izvršavanja koda zadatka 12: 17](#_Toc201530371)

[ZADATAK 13: Brojanje riječi u tekstu iz fajla 18](#_Toc201530372)

[Kod zadatka 13: 18](#_Toc201530373)

[Primjer izvršavanja koda zadatka 13: 19](#_Toc201530374)

[ZADATAK 14: Statistička analiza teksta iz fajla 19](#_Toc201530375)

[Kod zadatka 14: 19](#_Toc201530376)

[Primjer izvršavanja koda zadatka 14: 20](#_Toc201530377)

[15: Validacija korisničkog unosa - ime, email i lozinka 20](#_Toc201530378)

[Kod zadatka 15: 21](#_Toc201530379)

[Primjer izvršavanja koda zadatka 15: 22](#_Toc201530380)

# ZADATAK 1: Dijeljenje dva broja

**Opis zadatka:**

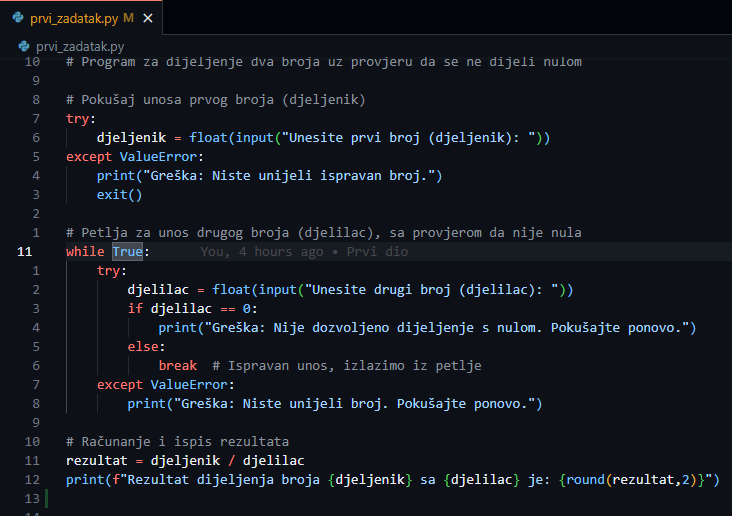
Napisati program koji od korisnika traži unos dva broja. Prvi broj predstavlja djeljenik, a drugi broj djelilac. Program mora izvršiti operaciju dijeljenja, uz obaveznu provjeru da djelilac nije nula. Ukoliko korisnik unese nulu kao djelilac, program treba prikazati poruku greške i tražiti ponovni unos dok se ne unese ispravan broj različit od nule.

Nakon što su oba broja uspješno unesena, program računa rezultat i prikazuje ga korisniku.

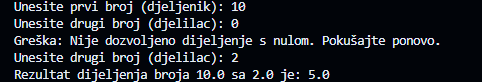
**Korištene funkcije i strukture:**

* **input()** - za unos podataka s tastature
* **float()** - za konverziju unosa u decimalni broj
* **while petlja** - za ponavljanje unosa dok korisnik ne unese validan broj
* **try-except blok** - za hvatanje grešaka prilikom konverzije unosa (npr. unos slova umjesto broja)
* **Operator / -** za dijeljenje dva broja
* **round()** - za zaokruživanje rezultata na dvije decimale radi čitljivosti

## Kod zadatka 1:



## Primjer izvršavanja koda zadatka 1:



# ZADATAK 2: Provjera parnog broja

**Opis zadatka:**

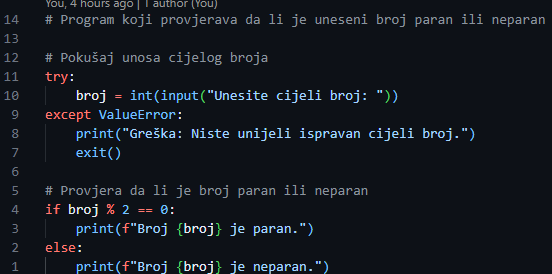
Napisati program koji traži od korisnika da unese cijeli broj. Program treba provjeriti da li je uneseni broj paran ili neparan, te ispisati odgovarajuću poruku.

Potrebno je osigurati da korisnik unese ispravan (validan) cijeli broj. U slučaju da korisnik unese vrijednost koja nije cijeli broj, program treba prikazati poruku o grešci i prekinuti izvođenje.

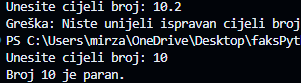
**Korištene funkcije i strukture:**

* **input()** - za unos podataka s tastature
* **int()** - za konverziju unosa u cijeli broj
* **try-except blok** - za hvatanje grešaka prilikom konverzije unosa
* **Operator %** - modulo operator, koristi se za provjeru ostatka pri dijeljenju

## Kod zadatka 2:



## Primjer izvršavanja koda zadatka 2:



# ZADATAK 3: Brojanje pojavljivanja svakog karaktera u stringu

**Opis zadatka:**

Napisati program koji omogućava korisniku unos proizvoljnog stringa (teksta). Program treba prebrojati koliko puta se svaki pojedinačni karakter (slovo, broj, znak ili razmak) pojavljuje u unesenom tekstu i na kraju ispisati rezultat.

Primjer: Ako korisnik unese tekst "banana", program treba ispisati koliko puta se slova 'a', 'p', 'l', 'e' pojavljuju.

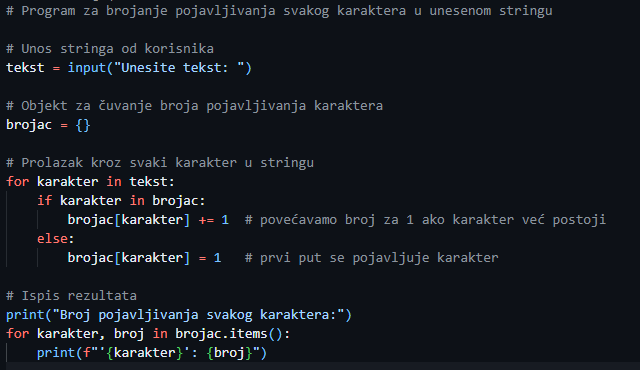
**Korištene funkcije i strukture:**

**input()** - za unos stringa

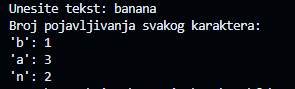
**for petlja** - za prolazak kroz svaki karakter u stringu

**items()** - za iteraciju kroz parove ključ-vrijednost u rječniku

## Kod zadatka 3:



## Primjer izvršavanja koda zadatka 3:



# ZADATAK 4: Prebrojavanje praznih mjesta, brojeva, slova i specijalnih karaktera u unesenom stringu

**Opis zadatka:**

Napisati program koji traži od korisnika da unese proizvoljan tekst (string). Program treba prebrojati i ispisati koliko se u tom tekstu nalazi:

* praznih mjesta (razmaka),
* cifara (brojeva 0-9),
* slova (velikih i malih),
* specijalnih karaktera (svi ostali znakovi koji nisu ni slovo, ni broj, ni prazno mjesto).

**Korištene funkcije i strukture:**

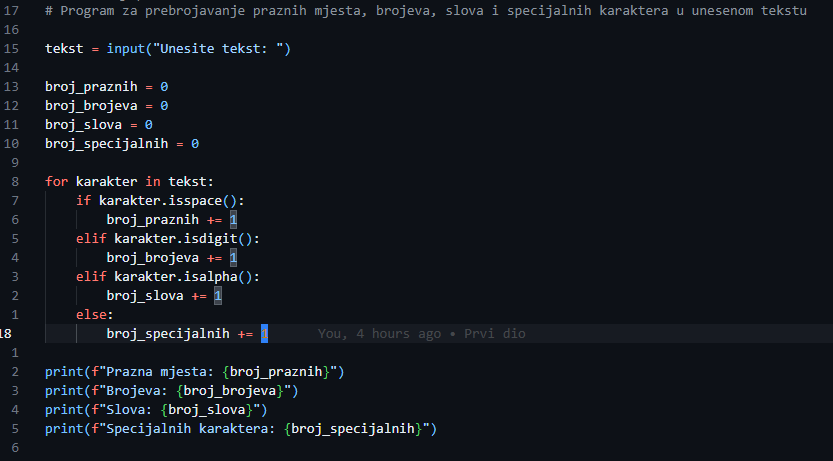
* **Input()** - za unos stringa
* **for petlja** - za iteraciju kroz svaki karakter u tekstu

Metode stringova:

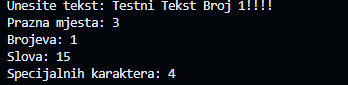
* **.isspace()** - provjerava da li je karakter prazno mjesto
* **.isdigit()** - provjerava da li je karakter broj
* **.isalpha()** - provjerava da li je karakter slovo

Ovaj zadatak pokazuje korištenje metoda za provjeru tipa karaktera i jednostavnu kategorizaciju znakova unutar stringa.

## Kod zadatka 4:



## Primjer izvršavanja koda zadatka 4:



# ZADATAK 5: Izračunavanje faktorijela broja korištenjem while petlje

**Opis zadatka:**

Napisati program koji traži od korisnika unos jednog pozitivnog cijelog broja, a zatim izračunava i ispisuje njegov faktorijel. Faktorijel broja n je proizvod svih pozitivnih cijelih brojeva od 1 do n. Na primjer, 5! = 5 × 4 × 3 × 2 × 1 = 120.

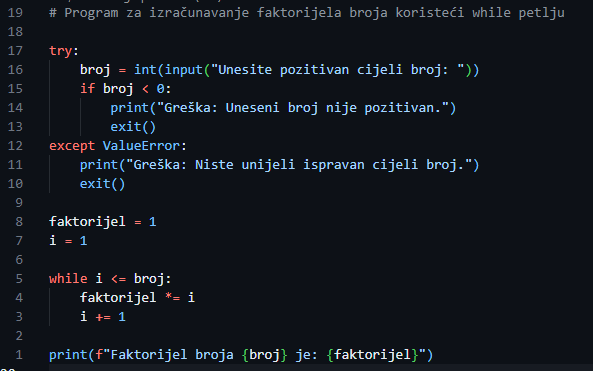
Program mora provjeriti da je uneseni broj pozitivan cijeli broj, u slučaju neispravnog unosa (negativan broj ili ne-cijeli broj) ispisati poruku o grešci i prekinuti program,

**Korištene funkcije i strukture:**

* **input()** - unos podataka
* **int()** - konverzija u cijeli broj
* **try-except blok** - za hvatanje greške prilikom konverzije
* **while petlja** - za iterativni izračun faktorijela

Korištenjem while petlje iterativno množimo sve brojeve od 1 do unesenog broja kako bismo dobili faktorijel.

## Kod zadatka 5:



## Primjer izvršavanja koda zadatka 5:



# ZADATAK 6: Brojanje samoglasnika u unesenoj rečenici

**Opis zadatka:**

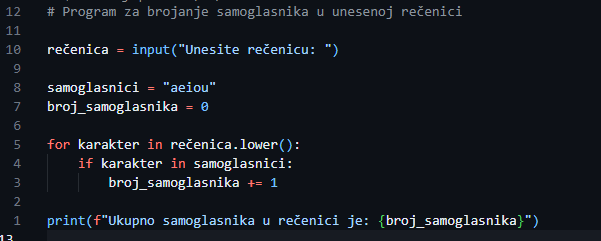
Napisati program koji traži od korisnika da unese rečenicu (string). Program treba prebrojati koliko se puta u toj rečenici pojavljuju samoglasnici. Samoglasnici su slova: a, e, i, o, u (uključujući i velika i mala slova).Na kraju, program treba ispisati ukupan broj samoglasnika u rečenici.

**Korištene funkcije i strukture:**

* **input()** - za unos stringa
* **for petlja** - za prolazak kroz svaki karakter u stringu
* **.lower()** - za pretvaranje karaktera u malo slovo, radi lakše provjere
* **if** - za provjeru da li je karakter samoglasnik

Program pretvara sve karaktere u mala slova kako bi provjera bila neosjetljiva na velika i mala slova, što olakšava prepoznavanje samoglasnika.

## Kod zadatka 6:



## Primjer izvršavanja koda zadatka 6:



# ZADATAK 7: Provjera da li je unesena riječ palindrom

**Opis zadatka:**

Napisati program koji traži od korisnika da unese jednu riječ. Program treba provjeriti da li je ta riječ palindrom, odnosno da li se riječ isto čita s lijeva na desno i s desna na lijevo.

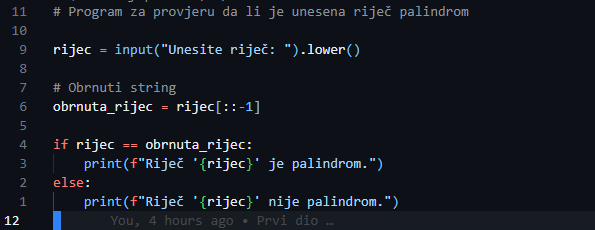
Primjer palindroma: "radar", "ana", "level".

Program treba ignorisati velika i mala slova prilikom provjere. Na kraju, program ispisuje odgovarajuću poruku da li je riječ palindrom ili nije.

**Korištene funkcije i strukture:**

* **input()** - za unos riječi
* **.lower()** - za pretvaranje riječi u mala slova
* **Slicing sa korakom -1** - za dobijanje obrnutog stringa
* **if** - za poređenje originalne i obrnute riječi

## Kod zadatka 7:



## Primjer izvršavanja koda zadatka 7:



# ZADATAK 8: Izračunavanje prosječne ocjene studenta

**Opis zadatka:**

Napisati program koji traži od korisnika da unese broj ocjena koje želi unijeti. Zatim program traži unos svih ocjena pojedinačno. Na kraju, program računa i ispisuje prosječnu ocjenu.

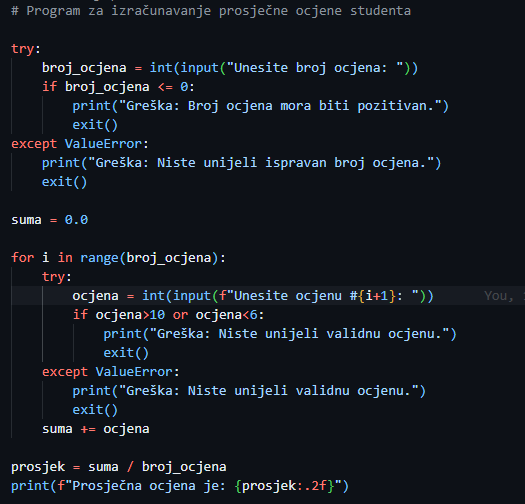
Program treba provjeriti da je broj ocjena pozitivan cijeli broj, tražiti unos ocjena cijelih brojeva između 6 i 10. Izračunati prosjek svih unesenih ocjena i ispisati ga sa dvije decimale.U slučaju neispravnog unosa, program treba ispisati odgovarajuću poruku i prekinuti rad.

**Korištene funkcije i strukture:**

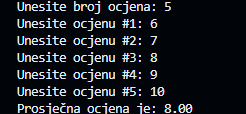
* **input()** - za unos podataka
* **int() i float()** - za konverziju unosa
* **try-except blok** - za hvatanje grešaka pri konverziji
* **for petlju** - za unos ocjena i sumiranje
* **:.2f** - za ispis rezultata sa dvije decimale

Ovaj zadatak pokazuje unos više podataka, njihovu validaciju, te korištenje petlje i aritmetičkih operacija za izračun prosjeka.

## Kod zadatka 8:



## Primjer izvršavanja koda zadatka 8:



# ZADATAK 9: Generisanje Fibonačijevog niza

**Opis zadatka:**

Napisati program koji traži od korisnika da unese broj n i zatim generiše i ispiše prvih n članova Fibonačijevog niza.

Fibonačijev niz je niz brojeva gdje je svaki član jednak zbiru prethodna dva, počevši od 0 i 1. Prvih nekoliko brojeva u nizu su: 0, 1, 1, 2, 3, 5, 8, ...

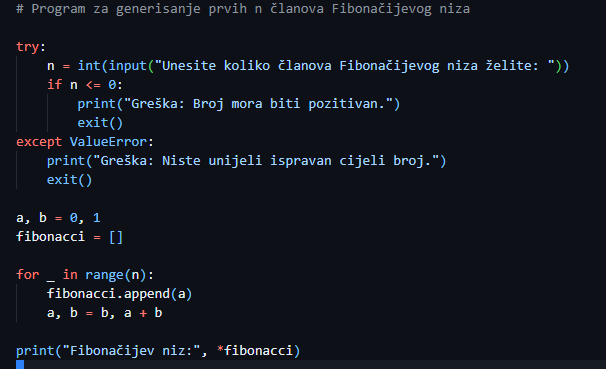
Program treba provjeriti da je n pozitivan cijeli broj,u slučaju neispravnog unosa ispisati poruku o grešci i prekinuti program, ispisati niz brojeva u jednom redu odvojenih razmakom.

**Korištene funkcije i strukture:**

* **input()** - za unos podataka
* **int()** - za konverziju unosa
* **try-except blok** - za hvatanje greške prilikom konverzije
* **for** - za generisanje niza

Ovaj zadatak prikazuje rad sa petljom i listama, te osnovnu implementaciju poznatog niza u matematici i programiranju.

## Kod zadatka 9:



## Primjer izvršavanja koda zadatka 9:



# ZADATAK 10: Kalkulator

**Opis zadatka:**

Napisati program koji predstavlja jednostavan kalkulator sa sljedećim opcijama u meniju: sabiranje, oduzimanje, množenje, dijeljenje i izlaz iz programa.

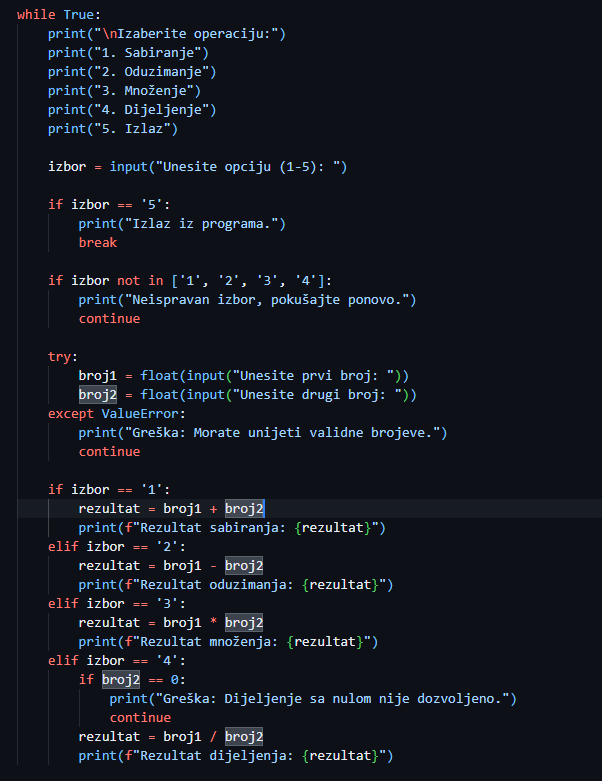
Program treba prikazati meni i tražiti od korisnika da izabere opciju, za opcije 1 do 4 tražiti unos dva broja, izvršiti odgovarajuću aritmetičku operaciju, za dijeljenje provjeriti da drugi broj nije 0 (kako se ne bi dijelilo sa nulom), ispisati rezultat operacije, omogućiti korisniku da ponovo bira opciju dok ne izabere izlaz (opcija 5).

**Korištene funkcije i strukture:**

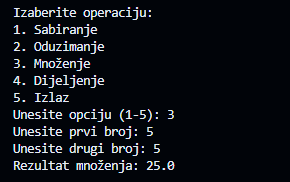
* **input()** - za unos podataka,
* **int() i float() -** za konverziju unosa,
* **while True** - za beskonačnu petlju menija,
* **if-elif-else** - struktura za izbor opcija,
* **try-except** - za hvatanje pogrešaka pri unosu,
* **break** - za izlazak iz petlje.

Program koristi beskonačnu petlju da bi omogućio višestruke operacije dok korisnik ne odluči izaći. Svaki korak unosa i operacije je provjeren da bi se izbjegle greške.

## Kod zadatka 10:

****

## Primjer izvršavanja koda zadatka 10:

****

# ZADATAK 11: Pogađanja broja

**Opis zadatka:**

Napisati program koji nasumično generiše cijeli broj između 1 i 100, a korisnik treba da pogađa taj broj. Nakon svakog pokušaja, program treba da korisniku da povratnu informaciju:

**"Manje"** – ako je uneseni broj veći od traženog broja,

**"Veće"** – ako je uneseni broj manji od traženog broja,

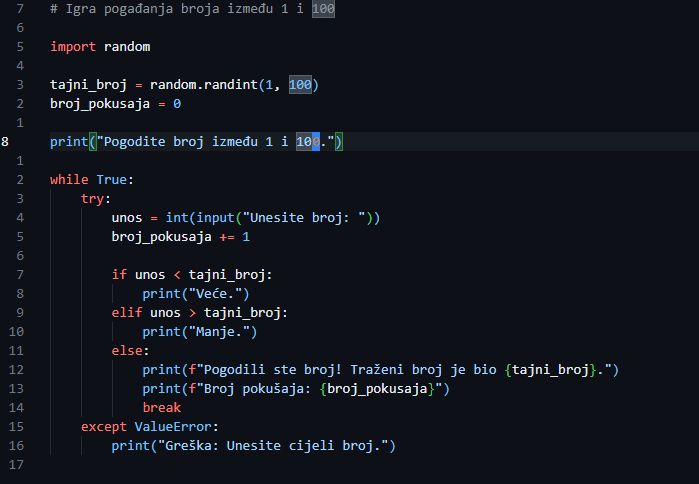
**"Pogodili ste broj!"** – ako je broj pogođen.

Program treba nasumično generisati broj iz opsega 1–100, omogućiti korisniku da unosi brojeve dok ne pogodi, brojati koliko je pokušaja bilo potrebno, na kraju ispisati broj pokušaja.

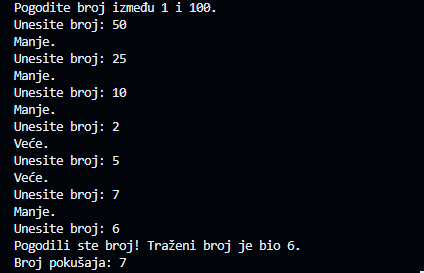
**Korištene funkcije i strukture:**

* **import random** – za generisanje nasumičnog broja
* **random.randint(1, 100)** – nasumični broj između 1 i 100
* **while petlja** – za ponavljanje pokušaja dok se broj ne pogodi
* **try-except** – za provjeru validnog unosa

## Kod zadatka 11:



## Primjer izvršavanja koda zadatka 11:



# ZADATAK 12: Rad sa fajlom - zapis i čitanje

**Opis zadatka:**

Napisati program koji traži od korisnika da unese nekoliko redova teksta (npr. tri rečenice), zapiše uneseni tekst u tekstualni fajl nazvan “dvanaesti.txt", nakon toga otvori isti fajl, pročita sadržaj i ispiše ga na ekranu.

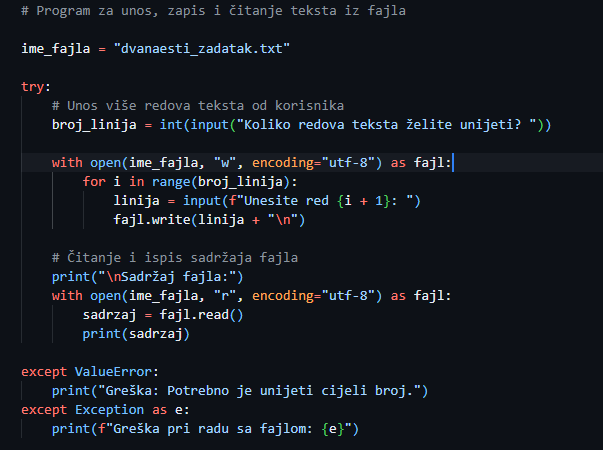
Program treba svaki korisnikov unos (linija) zapisuje se kao novi red u fajlu, koristiti with blok za rad sa fajlom, te rukovati eventualnim greškama prilikom upisa i čitanja fajla.

**Korištene funkcije i strukture:**

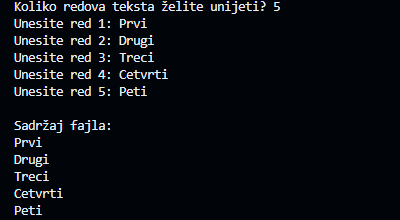
* **input()** — za unos teksta od strane korisnika
* **open()** — za otvaranje fajla (pisanje i čitanje)
* **with** — za automatsko zatvaranje fajla
* **try-except** — za hvatanje eventualnih grešaka
* **for petlja**— za unos više linija

Ovaj zadatak je koristan za demonstraciju osnovnog rada sa fajlovima u Pythonu, zapisivanje i čitanje sadržaja. Koristi se with blok koji osigurava da se fajl ispravno zatvori čak i ako dođe do greške.

## Kod zadatka 12:



## Primjer izvršavanja koda zadatka 12:



# ZADATAK 13: Brojanje riječi u tekstu iz fajla

**Opis zadatka:**

Napisati program koji otvara tekstualni fajl nazvan "trinaesti\_zadatak.txt", čita sav sadržaj iz fajla, prebrojava koliko riječi ima ukupno u tom tekstu, ispisuje broj riječi na ekranu. Ako fajl ne postoji ili dođe do greške prilikom čitanja, program treba ispisati odgovarajuću poruku o grešci.

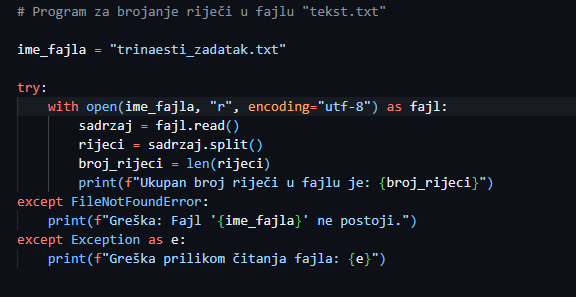
Program more koristiti with blok za čitanje fajla, pretpostaviti da su riječi razdvojene razmacima, koristiti split() metodu za izdvajanje riječi iz teksta i rukovati greškom ako fajl ne postoji.

**Korištene funkcije i metode:**

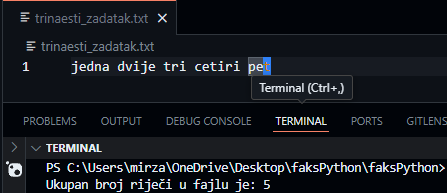
* **open()** – za otvaranje fajla
* **read()** – za čitanje kompletnog sadržaja fajla
* **split()** – za podjelu stringa na riječi
* **try-except** – za hvatanje grešaka
* **len()** – za brojanje riječi

Ovaj zadatak proširuje rad sa fajlovima analizom teksta. Program koristi .split() bez parametara, što automatski odvaja riječi po razmacima i novim redovima, što je dovoljno za osnovnu obradu teksta.

## Kod zadatka 13:



## Primjer izvršavanja koda zadatka 13:



# ZADATAK 14: Statistička analiza teksta iz fajla

**Opis zadatka:**

Napisati program koji iz fajla "cetrnaesti\_zadatak.txt", čita sav tekst iz fajla, broji koliko puta se pojavljuje svaka pojedinačna riječ, ispisuje 10 najčešćih riječi zajedno sa brojem njihovih pojavljivanja, ignoriše velika i mala slova (radi lakšeg prebrojavanja).

Program mora koristiti collections, counter za brojanje riječi, tekst prije analize pretvoriti u mala slova ukloniti znakove interpunkcije (tačke, zareze itd. radi lakšeg prebrojavnja) i rukovati greškom ako fajl ne postoji.

## Kod zadatka 14:



## Primjer izvršavanja koda zadatka 14:



# 15: Validacija korisničkog unosa - ime, email i lozinka

**Opis zadatka**:  
Napisati program koji traži od korisnika da unese ime i prezime, email adresu i lozinku. Program treba validirati svaki od ovih unosa prema zadatim pravilima koristeći regex. Ovo simulira backend validaciju podataka koja se često koristi u web aplikacijama.

**Validacija imena:**

Mora sadržavati samo slova i razmake (bez brojeva ili specijalnih znakova) i mora imati barem dvije riječi (ime i prezime).

**Validacija email adrese:**

Mora sadržavati znak '@' i mora završavati na jednu od dozvoljenih domena: .com, .ba, .org ili .edu.

**Validacija lozinke:**

Mora imati najmanje 8 karaktera i mora sadržavati bar jedno veliko slovo, jedno malo slovo, jedan broj i jedan specijalni znak.

**Korišteni regex izrazi i njihovo značenje:**

**Za ime**: ^[A-Za-zŠĐČĆŽšđčćž ]+$

* Ovaj izraz dozvoljava samo velika i mala slova, uključujući slova sa, razmake. ^ i $ označavaju početak i kraj stringa, što znači da cijeli unos mora biti u ovom formatu.

**Za email**: ^[\w\.-]+@[\w\.-]+\.(com|ba|org|edu)$

* Ovaj izraz traži da email počinje sa slovima, brojevima, tačkama ili crtama ([\w\.-]+), zatim ima znak @, poslije opet niz znakova i na kraju završava jednom od dozvoljenih domena. $ garantuje kraj stringa, dakle ništa ne može slijediti nakon domena.

**Za lozinku** se koriste odvojene regex provjere:

* Veliko slovo: [A-Z]
* Malo slovo: [a-z]
* Broj: [0-9]
* Specijalni znak: [!@#$%^&\*(),.?":{}|<>] (može biti prilagođeno)

Svaka od ovih provjera se radi posebno kako bi se osiguralo da lozinka zadovoljava sve kriterije.

**Korištene funkcije i strukture:**

* **input()** — za unos podataka od korisnika
* **re.match()** i **re.search()** — za provjeru unosa preko regularnih izraza
* **len()** — za provjeru dužine lozinke
* **if uslovi** — za logiku validacije i ispis poruka korisniku

## Kod zadatka 15:



## Primjer izvršavanja koda zadatka 15:

