

PROGRAMIRANJE

ZBIRKA ZADATAKA

Ishod učenja 2

2023-2024.

Zadatak 1

Napišite program koji učitava pet imena od korisnika u listu, a zatim učitava i broj N. Neka program ispiše sva imena iz liste čiji je broj znakova manji od N. Na primjer, ako korisnik unese "Anna", "Mary", "Peter", "Bart" i "Anastasia" i N=5, program bi trebao prikazati:

Anna
Mary
Bart

Moguće rješenje:

```
imena = []
for i in range(5):
    ime = input("Upišite ime: ")
    imena.append(ime)

n = int(input("Upišite n: "))

for i in range(len(imena)):
    if len(imena[i]) < n:
        print(imena[i])
```

Zadatak 2

Napišite program koji korisniku omogućuje čuvanje kratica i njihovih objašnjenja (primjerice, može spremi kraticu „HTTP“ s objašnjenjem „Hyper Text Transfer Protocol“). Omogućite korisniku sljedeće:

- Unos nove kratice i objašnjenja. Ako je kratica već prije unesena, obavijestite korisnika o tome i pitajte ga želi li zaista promijeniti objašnjenje.
- Ispis objašnjenja prema unesenoj kratici.
- Uklanjanje kratice i objašnjenja.
- Kraj rada.

Moguće rješenje:

```
baza_podataka = {}

print("A = unos novog zapisa")
print("B = traženje")
print("C = uklanjanje")
print("Q = kraj")

while True:
    akcija = input("> ")
    if akcija == "A":
        kratica = input("Upišite kraticu: ")
        objasnjenje = input("Upišite objašnjenje: ")
        if kratica in baza_podataka:
            if input("Kratica već postoji, želite li zamijeniti objašnjenje (da, ne): ") == "da":
                baza_podataka[kratica] = objasnjenje # update
            else:
                baza_podataka[kratica] = objasnjenje # insert
        elif akcija == "B":
            kratica = input("Upišite kraticu: ")
            if kratica in baza_podataka:
                print(baza_podataka[kratica])
            else:
                print("Nepostojeća kratica")
        elif akcija == "C":
            kratica = input("Upišite kraticu: ")
            if kratica in baza_podataka:
                del baza_podataka[kratica]
            else:
                print("Nepostojeća kratica")
        elif akcija == "Q":
            break
```

Zadatak 3

Napišite program koji generira deset slučajnih brojeva između 1 i 100 i smješta ih u listu. Nakon toga izračunajte prosjek tih brojeva. Nakon toga kopirajte sve brojeve veće ili jednake prosjeku u drugu listu. Na kraju ispišite sve brojeve iz prve liste, zatim prosjek, a zatim sve brojeve iz druge liste.

Moguće rješenje:

```
import random

brojevi = []

# Prvo: napunimo listu slučajnim brojevima
for i in range(10):
    slucajni = random.randint(1, 100)
    brojevi.append(slucajni)

# Drugo: izracunamo prosjek
sum = 0.0;
for i in range(len(brojevi)):
    sum += brojevi[i]

average = sum / 10;

# Treće: kopiramo u novu listu
veci = []

for i in range(len(brojevi)):
    if brojevi[i] >= average:
        veci.append(brojevi[i])

# Četvrto: ispisemo
print("Brojevi iz prve liste: ", end="")
for i in range(len(brojevi)):
    print(brojevi[i], end=" ")

print("\nProsjek je:", average)
print("Brojevi iz druge liste: ", end="")
for i in range(len(veci)):
    print(veci[i], end=" ")
```

Zadatak 4

Napišite program koji generira stotinu slučajnih brojeva između 1 i 100 i smješta ih u listu. Nakon toga, tražite korisnika da upiše broj, a vi izbrišite sve brojeve iz liste koji su veći od tog broja. Ponavljajte dok to korisnik želi. Na kraju, ispišite što je preostalo u listi.

Moguće rješenje:

```
import random

brojevi = []

for i in range(100):
    brojevi.append(random.randint(1, 100))

while True:
    broj = int(input("Upišite maksimalni broj koji smije ostati u listi: "))

    for i in range(len(brojevi) - 1, -1, -1):
        if brojevi[i] > broj:
            del brojevi[i]

    if input("Želite li završiti (da, ne): ") == "da":
        break

print(brojevi)
```

Zadatak 5

Napišite program koji priprema listu s tri naziva filma. Ispišite sve filmove od kraja liste do početka liste, ali tako da naziv svakog filma ispišete od kraja do početka. Na primjer, ako su filmovi u listi "Matrix", "Star Wars", "Finding Nemo", tada bi ispis trebao biti:

```
omeN gnidniF  
sraW ratS  
xirtaM
```

Moguće rješenje:

```
filmovi = [ "Matrix", "Star Wars", "Finding Nemo" ]  
  
for i in range(len(filmovi) - 1, -1, -1):  
    trenutni_film = filmovi[i]  
  
    for j in range(len(trenutni_film) - 1, -1, -1):  
        print(trenutni_film[j], end="")  
  
    print()
```

Zadatak 6

Napišite program koji priprema listu s nekoliko URL-ova (primjerice, <http://www.google.com> i slično). Nakon toga, prođite kroz listu i svaku adresu dodajte u rječnik ili pod ključem „sigurne“ ili pod ključem „nesigurne“ (sigurne adrese su HTTPS). Pošto i sigurnih i nesigurnih adresa može biti više, razmislite što će biti vrijednost u rječniku pod oba ključa. Nakon toga, tražite korisnika da upiše želi li prikazati sigurne ili nesigurne adrese pa tako i napravite.

Moguće rješenje:

```
lista_adresa = [
    "https://www.google.hr/",
    "http://www.bla.com/",
    "https://www.linkedin.com/",
    "https://www.orcid.com/",
    "https://scholar.google.com/"
]

adrese = {
    "sigurne": [],
    "nesigurne": []
}

for i in range(len(lista_adresa)):
    if lista_adresa[i][4] == ":":
        adrese["nesigurne"].append(lista_adresa[i])
    else:
        adrese["sigurne"].append(lista_adresa[i])

trazim = input("Upišite kakve adrese želite ispisati, sigurne ili nesigurne: ")
if trazim in adrese:
    for a in adrese[trazim]:
        print(a)
else:
    print("Neispravna vrsta adresa")
```

Zadatak 7

Napišite program koji sadrži dvije liste. Neka jedna lista sadrži sljedeće pridjeve: Amusing, Awesome, Beautiful, Good i Happy. Neka druga lista sadrži sljedeće imenice: dog, cat, bird i cloud. Neka vaš program ispiše jedan slučajni pridjev s jednom slučajnom imenicom. Ponavljajte onoliko dugo koliko korisnik želi. Primjer ispisa programa može biti:

```
Good dog
Zelite li ispisati jos jednu frazu (da, ne): da
Beautiful cat
Zelite li ispisati jos jednu frazu (da, ne): da
Good cloud
Zelite li ispisati jos jednu frazu (da, ne): da
Good dog
Zelite li ispisati jos jednu frazu (da, ne): da
Amusing dog
Zelite li ispisati jos jednu frazu (da, ne): ne
```

Moguće rješenje:

```
import random

pridjevi = [ "Amusing", "Awesome", "Beautiful", "Good", "Happy" ]
imenice = [ "dog", "cat", "bird", "cloud" ]

while True:
    # Postoji 5 pridjeva, ispravni indeksi su 0, 1, 2, 3 i 4
    slucajni_indeks_pridjevi = random.randint(0, 4)

    # Postoji 4 imenice, ispravni indeksi su 0, 1, 2 i 3
    slucajni_indeksi_imenice = random.randint(0, 3)

    # Print
    print(pridjevi[slucajni_indeks_pridjevi], imenice[slucajni_indeksi_imenice])

    # Pitamo korisnika zeli li dalje
    if input("Zelite li ispisati jos jednu frazu (da, ne): ") == "ne":
        break
```


Zadatak 8

Napišite program koji od korisnika učitava rečenicu. Nakon toga, neka program u rječnik smjesti sve znakove iz rečenice i broj pojavljivanja svakog znaka u rečenici. Na kraju, ispišite sve znakove i broj pojavljivanja svakog znaka.

Moguće rješenje:

```
pojavljivanje = {}

recenica = input("Upišite rečenicu: ")
for i in range(len(recenica)):
    trenutni_znak = recenica[i]
    if trenutni_znak in pojavljivanje: # Već postoji
        pojavljivanje[trenutni_znak] += 1
    else: # Prvi put nalazim ovaj znak
        pojavljivanje[trenutni_znak] = 1

for znak, broj_pojavljivanja in pojavljivanje.items():
    print("Znak", znak, "se pojavljuje", broj_pojavljivanja, "puta")
```

Zadatak 9

Napišite program koji korisniku nudi sljedeće opcije, sve dok korisnik to želi:

- Generiranje pet slučajnih godina od 1995. do 2032.
- Prikaz svih generiranih godina.

Primjer prikaza programa:

```
A = Generiranje slučajnih godina
B = Ispis svih generiranih godina
X = Izlaz
> A
> A
> B
2006 2025 2024 2024 2001 2027 2020 2009 1997 1998
> A
> B
2006 2025 2024 2024 2001 2027 2020 2009 1997 1998 2021 2005 1996 2015 2005
> X
```

Moguće rješenje:

```
import random

print("A = Generiranje slučajnih godina")
print("B = Ispis svih generiranih godina")
print("X = Izlaz")

generirane_godine = []

while True:
    izbor = input("> ")

    if izbor == "A":
        for i in range(5):
            slucajna_godina = random.randint(1995, 2032)
            generirane_godine.append(slucajna_godina)
    elif izbor == "B":
        for i in range(len(generirane_godine)):
            print(generirane_godine[i], end=" ")
        print()
    elif izbor == "X":
        break
```

Zadatak 10

Napišite program koji simulira rad DNS-a. Pripravite najprikladniji kontejner i u njega dodajte nekoliko elemenata. Nakon toga, tražite od korisnika da upiše neko domensko ime (primjerice. www.amazon.com), a vi mu u kontejneru pronađite i ispišite IP adresu (primjerice, 192.0.2.44).

Moguće rješenje:

```
dns_baza = {
    "www.amazon.com": "192.0.2.44",
    "www.microsoft.com": "2.18.69.217",
    "azure.microsoft.com": "104.103.104.184",
    "www.plezanje.net": "104.21.38.92",
    "algebra.hr": "178.218.162.131",
}

upit = input("Upišite naziv hosta čiju IP adresu želite saznati: ")

if upit in dns_baza:
    print(dns_baza[upit])
else:
    print("Traženi host nije u bazi, pitajte ishodišni DNS poslužitelj")
```

Zadatak 11

Napišite program koji simulira miješanje pjesama. Prvo stvorite listu s 5 imena pjesama. Zatim zamijenite dvije slučajne pjesme u listi i nastavite tako mijenjati slučajne pjesme 100 puta. Na kraju, prikažite sve pjesme iz liste.

Moguće rješenje:

```
import random

pjesme = [ "Fade to black", "A moment of clarity", "One", "Enter sandman", "Story to tell" ]

preostalo = 100

while preostalo > 0:
    ri1 = random.randint(0, 4)
    ri2 = random.randint(0, 4)

    if ri1 != ri2:
        # Zamijenimo ih
        temp = pjesme[ri1]
        pjesme[ri1] = pjesme[ri2]
        pjesme[ri2] = temp

        preostalo -= 1

for i in range(len(pjesme)):
    print(pjesme[i])
```

Zadatak 12

Napišite program i u njemu pripremite dva rječnika sa po 5 zapisa u svakom. Neka oba rječnika imaju cijeli broj kao ključ te jedno slovo kao vrijednost. Nakon toga, spojite dva rječnika u treći tako da prvo prekopirate sve zapise iz prvog rječnika, a zatim iz drugog. U slučaju da naiđete na već postojeći ključ, pitajte korisnika želi li pregaziti već postojeću vrijednost ili preskočiti taj zapis. Na kraju, ispišite treći rječnik u obliku: ključ = vrijednost

Moguće rješenje:

```
prvi = {
    1: "a",
    2 : "e",
    3 : "i",
    4 : "o",
    5 : "u",
}

drugi = {
    4: "r",
    5 : "t",
    6 : "z",
    7 : "k",
    8 : "l",
}

treći = {}

for k, v in prvi.items() :
    treći[k] = v

    for k, v in drugi.items() :
        if k in treći :
if input("Ključ " + str(k) + " već postoji, želite li ga prepisati (da, ne): ") ==
"da" :
    treći[k] = v
else :
    treći[k] = v

    for k, v in treći.items() :
        print(k, "=", v)
```

Zadatak 13

Napišite program koji simulira mehanizam pretvaranja RNA sekvence u proteine. Neka program traži od korisnika da unese RNA sekvencu (koju čine niz slova U, C, G i A). Zatim ispišite niz proteina koji će biti kreiran, na temelju sljedećih pravila prevođenja:

- AUG: početak – ignorirajte sva prethodna slova.
- UUU: će kreirati fenilalanin
- CAA: će kreirati glutamin
- GCA: će kreirati alanin
- ACC: će kreirati treonin
- UAA, UAG, UGA: kraj – kraj procesiranja, ignorirajte sva sljedeća slova

Ako u sekvenci postoji početak, možete biti sigurni da će kraj također biti prisutan, te da ću između biti ispravne kombinacije slova. Primjer valjanih ulaznih RNA sekvenci su:

AUGACCUAA – prevodi se kao početak, treonin, kraj

AUGGCAACCUGA – prevodi se kao početak, alanin, treonin, kraj

MEOWAUGACCUAAWOOF – MEOW i WOOF se zanemaruju pa se prevodi kao početak, treonin, kraj

MEOWMOEWWOOF – prevodi se kao "nije pronađen AUG".

Moguće rješenje:

```
rna = input("Unesite RNA sekvencu: ")

# Trazimo AUG, moze biti bilo gdje u stringu (ili cak nigdje)
pocetni_indeks = -1
for i in range(len(rna) - 2):
    if rna[i] == 'A' and rna[i + 1] == 'U' and rna[i + 2] == 'G':
        pocetni_indeks = i
        break;

# Nastavljamo samo ako smo nasli AUG
if pocetni_indeks > -1:
    print("pocetak")

    for i in range(pocetni_indeks, len(rna) - 2, 3):
        if rna[i] == 'U' and rna[i + 1] == 'U' and rna[i + 2] == 'U':
            print("fenilalanin")
        elif rna[i] == 'C' and rna[i + 1] == 'A' and rna[i + 2] == 'A':
            print("glutamin")
        elif rna[i] == 'G' and rna[i + 1] == 'C' and rna[i + 2] == 'A':
            print("alanin")
        elif rna[i] == 'A' and rna[i + 1] == 'C' and rna[i + 2] == 'C':
            print("treonin")
        elif rna[i] == 'U' and rna[i + 1] == 'A' and rna[i + 2] == 'A' or rna[i] ==
'U' and rna[i + 1] == 'A' and rna[i + 2] == 'G' or rna[i] == 'U' and rna[i + 1] ==
'G' and rna[i + 2] == 'A':
            print("kraj")
            break;
    else:
        print("nije pronadjen AUG")
```

Zadatak 14

Napišite program koji koristi jednostavnu tehniku šifriranja. Napravite dvije sekvence: prva treba sadržavati sva mala slova od a-z, a druga također, ali obrnutim redoslijedom, od z-a. Zatim učitajte riječ od korisnika, pronađite svaki znak u prvoj sekvenci i ispišite znak na istom indeksu u drugoj sekvenci. Ako naidete na nepoznati znak, samo ga ispišite. Primjer rada programa:

Upisite recenicu koju zelite sifrirati: danas je lijep i sunčan dan
wzmzh qv orqvk r hfmxzm wzm

Moguće rješenje:

```
first = "abcdefghijklmnopqrstuvwxyz"
second = "zyxwvutsrqponmlkjihgfedcba"

# Ucitamo recenicu
recenica = input("Upisite recenicu koju zelite sifrirati: ")

# Sifriranje
for i in range(len(recenica)):
    trenutno_slovo = recenica[i]

    # Pronadjemo u prvoj sekvenci
    pronadjeno = False
    for j in range(len(first)):
        if first[j] == trenutno_slovo:
            # Pronadjeno na indeksu j, ispisemo iz druge sekvence
            print(second[j], end="")
            pronadjeno = True
            break

    # Ako nije pronadjeno u prvom polju, samo ga ispisemo
    if not pronadjeno:
        print(trenutno_slovo, end="")

print()
```

Zadatak 15

Napišite program koji stvara dvije liste: prva sadrži pet decimalnih brojeva, a druga pet boolova. Za svaki indeks u drugoj listi na kojem je True, uzmite broj iz prve liste na istom indeksu i kopirajte ga u novu listu. Na kraju ispišite sadržaj liste na sljedeći način: neka svaki element bude uvučen za sljedeći broj razmaka: indeksu u listi * 4. Na primjer:

```
12.8
   4.5
      12.001
```

Moguće rješenje:

```
brojevi = [ 12.8, 4.5, -3.4, 12.001, 7.6 ]
treba_kopirati = [ True, True, False, True, False ]
v = []

# Kopiramo sto treba
for i in range(len(brojevi)):
    if treba_kopirati[i] == True:
        v.append(brojevi[i])

for i in range(len(v)):
    # Uvlacenje
    for j in range(4 * i):
        print(" ", end="")

    print(v[i])
```


Zadatak 16

Napišite program koji stvara slučajne prodajne vrijednosti za svaki mjesec u godini. Neka svaka prodajna vrijednost bude u rasponu od 1.000 dolara do 50.000 dolara. Nakon toga, prikažite trakasti graf gdje u svakom retku prikazujete broj mjeseca i jedan znak '#' za svakih punih 1.000 dolara iz tog mjeseca. Također, na kraju svakog retka prikažite prodajnu vrijednost. Na primjer:

```
Mjesec 1: ##### (21154 USD)
Mjesec 2: ##### (31945 USD)
Mjesec 3: ##### (27940 USD)
Mjesec 4: ##### (10703 USD)
Mjesec 5: ##### (31751 USD)
Mjesec 6: ##### (6249 USD)
Mjesec 7: ##### (18194 USD)
Mjesec 8: ##### (14551 USD)
Mjesec 9: ### (3772 USD)
Mjesec 10: ##### (24096 USD)
Mjesec 11: ##### (17583 USD)
Mjesec 12: ##### (31486 USD)
```

Moguće rješenje:

```
import random

prodano = []

# Kreiranje slučajnih prodajnih vrijednosti
for i in range(12):
    slučajni_broj = random.randint(1000, 50000)
    prodano.append(slučajni_broj)

# Prikaz
for i in range(len(prodano)):
    print("Mjesec", i + 1, end=":\t")

    koliko_hasheva = int(prodano[i] / 1000)

    for j in range(koliko_hasheva):
        print("#", end="")

    print(" (" + str(prodano[i]) + " USD)")
```

Zadatak 17

Napišite program koji učitava 1D točke (to jest, cijele brojeve) dok to korisnik želi. Kada je unos gotov, ispišite taj 1D koordinatni sustav od 0 do 50, koristeći 'x' za označavanje korisnički definirane točke i '-' za označavanje praznog prostora. Primjer:

```
Upisite tocku: 50
Dodati jos jednu tocku (da/ne): da
Upisite tocku: 4
Dodati jos jednu tocku (da/ne): da
Upisite tocku: 40
Dodati jos jednu tocku (da/ne): ne
-----x-----x-----x
```

Moguće rješenje:

```
tocke = []

while True:
    tocka = int(input("Upisite tocku: "))
    tocke.append(tocka)

    if input("Dodati jos jednu tocku (da/ne): ") == "ne":
        break

for x in range(51):
    postoji = False

    for i in range(len(tocke)):
        if tocke[i] == x:
            print("x", end="")
            postoji = True
            break

    if not postoji:
        print("-", end="")

print()
```

Zadatak 18

Napišite program koji kreira tri liste cijelih brojeva: prva treba sadržavati 5 cijelih brojeva, druga također treba sadržavati 5 cijelih brojeva, a treća treba inicijalno biti prazna. Zatim kopirajte elemente iz prve u treću listu, a potom i iz druge u treću listu. Na taj način treća lista treba sadržavati sve vrijednosti iz prve dvije liste. Na kraju, prikažite sadržaj treće liste.

Moguće rješenje:

```
prva = [ 10, 20, 30, 40, 50 ]  
druga = [ 60, 70, 80, 90, 100 ]  
treca = []
```

```
for i in range(len(prva)):  
    treca.append(prva[i])
```

```
for i in range(len(druga)):  
    treca.append(druga[i])
```

```
for i in range(len(treca)):  
    print(treca[i], end=" ")
```

Zadatak 19

Napišite program koji učitava kategoričke vrijednosti (kao što su "vrlo loše", "loše", "dobro", "vrlo dobro" itd.) koliko to korisnik želi. Nakon toga, učitajte od korisnika jednu kategoričku vrijednost X i zatim prikažite sve prethodno učitane vrijednosti na sljedeći način: ako je vrijednost jednaka X, prikažite 1; u suprotnom, prikažite 0. Primjer programa:

```
Upisite kategoricku vrijednost: dobro
Dodati jos jednu (1=da, 0=ne): da
Upisite kategoricku vrijednost: lose
Dodati jos jednu (1=da, 0=ne): da
Upisite kategoricku vrijednost: dobro
Dodati jos jednu (1=da, 0=ne): da
Upisite kategoricku vrijednost: vrlo dobro
Dodati jos jednu (1=da, 0=ne): ne
Upisite kategoricku vrijednost X: dobro
1010
```

Moguće rješenje:

```
vrijednosti = []

while True:
    vrijednost = input("Upisite kategoricku vrijednost: ")
    vrijednosti.append(vrijednost)

    if input("Dodati jos jednu (da, ne): ") == "ne":
        break

trazi = input("Upisite kategoricku vrijednost X: ")

for i in range(len(vrijednosti)):
    if vrijednosti[i] == trazi:
        print("1", end="")
    else:
        print("0", end="")

print()
```

Zadatak 20

Napišite program za rad s jednostavnim upitnim jezikom (BSQL). Jezik se koristi za zadavanje upita nad bazom podataka: listom znakova. Neka ovaj jezik sadrži tri naredbe: SELECT, INSERT i UPDATE:

- SELECT – dohvaća sve vrijednosti iz liste i prikazuje ih.
- INSERT <vrijednost> – dodati vrijednost u listu.
- UPDATE <vrijednost> TO <nova-vrijednost> – promijeniti sve vrijednosti u listi na novu.

Iskoristite činjenicu da je vrijednost uvijek jedan znak i da će korisnik uvijek ispravno unijeti jednu od tri moguće naredbe. U beskonačnoj petlji dopustite korisniku da manipulira listom. Na primjer:

Koristite sljedeće naredbe:

```
SELECT
INSERT <vrijednost>
UPDATE <vrijednost> TO <nova-vrijednost>
> INSERT a
> INSERT b
> SELECT
a b
> INSERT c
> INSERT a
> SELECT
a b c a
> UPDATE a TO z
> SELECT
z b c z
...
```

Moguće rješenje:

```
baza_podataka = []

print("Koristite sljedeće naredbe:")
print("SELECT")
print("INSERT <vrijednost>")
print("UPDATE <vrijednost> TO <nova-vrijednost>")

while True:
    naredba = input("> ")

    if naredba == "SELECT":
        for i in range(len(baza_podataka)):
            print(baza_podataka[i], end=" ")
        print()
    elif naredba[0] == 'I': # insert
        nova_vrijednost = naredba[7]
        baza_podataka.append(nova_vrijednost)
    else: # update
        stara_vrijednost = naredba[7]
        nova_vrijednost = naredba[12]

        for i in range(len(baza_podataka)):
            if baza_podataka[i] == stara_vrijednost:
                baza_podataka[i] = nova_vrijednost
```

Zadatak 21

Napišite program koji simulira igru pogađanja riječi. Definirajte popis riječi napisanih malim slovima, a zatim odaberite slučajnu riječ. Dopustite korisniku da pogađa znakove dok ne pogodi sve. Nakon svakog unosa znaka, prikažite riječ tako da ispišete sve pogođene znakove, a nepogođene prikažite kao '_'. Ispišite broj pokušaja koji je bio potreban da se pogodi cijela riječ. Na primjer:

```
Pogodite slovo: a
_a_a_a_
Pogodite slovo: c
_a_a_ac
Pogodite slovo: f
_a_a_ac
Pogodite slovo: m
ma_a_ac
Pogodite slovo: g
maga_ac
Pogodite slovo: r
magarac
Pogodili ste u 6 pokušaja
```

Moguće rješenje:

```
import random

rijeci = [ "integral", "brokula", "magarac", "prevjes", "studeni" ]
odabrana_rijec = rijeci[random.randint(0, 4)]
pokusaji = []
broj_pokusaja = 0

while True:
    pokusaji.append(input("Pogodite slovo: "))
    broj_pokusaja += 1

    # Ispis rijeci
    pogodjeno_slova = 0
    for i in range(len(odabrana_rijec)):
        trenutno_slovo = odabrana_rijec[i]

        if trenutno_slovo in pokusaji:
            print(trenutno_slovo, end="")
            pogodjeno_slova += 1
        else:
            print("_", end="")
    print()

    # Sve pogodjeno?
    if pogodjeno_slova == len(odabrana_rijec):
        print("Pogodili ste u", broj_pokusaja, "pokusaja")
        break
```

Zadatak 22

Napišite program koji od korisnika traži da unese telefonski broj kao niz od 10 znamenki. Zatim ispišite formatiranu verziju istog broja. Nakon prve tri znamenke ispišite '/', a nakon sljedeće 4 znamenke ispišite '-', kao u primjeru:

Upisite telefonski broj: 0998902314

Formatirani telefonski broj: 099/8902-314

Moguće rješenje:

```
broj = input("Upisite telefonski broj: ")

print("Formatirani telefonski broj: ", end="")
for i in range(len(broj)):
    print(broj[i], end="")

    if i == 2:
        print("/", end="")
    elif (i == 6):
        print("-", end="")
```

Zadatak 23

Napišite program koji od korisnika učitava rečenicu i slovo koje treba pronaći. Zatim pronađite sva pojavljivanja tog slova u rečenici i ispišite njihove indekse. Ako nema nijednog pojavljivanja, ispišite odgovarajuću poruku. Na primjer:

```
Upisite recenicu: ovo je primjer jednostavne recenice
Upisite slovo koje zelite pronaci: e
Pronadjeno na indeksu: 5
Pronadjeno na indeksu: 12
Pronadjeno na indeksu: 16
Pronadjeno na indeksu: 25
Pronadjeno na indeksu: 28
Pronadjeno na indeksu: 30
Pronadjeno na indeksu: 34
```

Moguće rješenje:

```
recenica = input("Upisite recenicu: ")
trazi = input("Upisite slovo koje zelite pronaci: ")
pojavljivanja = 0

for i in range(len(recenica)):
    if recenica[i] == trazi:
        print("Pronadjeno na indeksu:", i, )
        pojavljivanja += 1

if pojavljivanja == 0:
    print("Nije pronadjeno")
```


Zadatak 24

Napišite program koji komprimira niz znakova. Kompresija se izvodi zamjenom svakog niza nula s količinom nula u nizu, navedenim u zagradama. Sve ostale znakove treba samo ispisati. Neka program traži od korisnika da unese niz znakova i onda mu ispiše komprimiranu verziju. Na primjer:

Upisite niz: 2000006300250000302090000000

Komprimirano: 2(5)63(2)25(4)3(1)2(1)9(7)

Moguće rješenje:

```
niz = input("Upisite niz: ")

brojim_nule = False
pojavljivanja = 0

print("Komprimirano: ", end="")
for i in range(len(niz)):
    if niz[i] == '0':
        if not brojim_nule:
            # Pronašli smo prvu nulu u sekvenci
            brojim_nule = True
            pojavljivanja += 1
        else:
            if brojim_nule:
                # Upravo smo prestali brojati nule
                brojim_nule = False
                print("(" + str(pojavljivanja) + ")", end="")
                pojavljivanja = 0

            print(niz[i], end="")

if pojavljivanja > 0:
    print("(" + str(pojavljivanja) + ")", end="")

print()
```

Zadatak 25

Napišite program koji kreira listu elemenata i provjerava je li taj niz elemenata strogo rastući (strogo rastući znači da je svaki element veći od prethodnika). Na primjer, ovaj niz je strogo rastući: 2 3 5 8 9 10 15 47, a ovaj nije: 2 3 5 4 9 10 15 47.

Moguće rješenje:

```
niz = [ 2, 4, 6, 8, 10, 11, 14, 16 ]

strogo_rastuci = True

for i in range(len(niz) - 1):
    if niz[i] >= niz[i + 1]:
        strogo_rastuci = False
        break

if strogo_rastuci:
    print("Niz je strogo rastuci")
else:
    print("Niz nije strogo rastuci")
```

Zadatak 26

Napišite program koji stvara dvije liste, jedna sadrži 8 imena, a druga 8 cijelih brojeva. Zatim prikažite imena iz prve liste uvučena onoliko razmaka koliko piše na istom indeksu u drugoj listi.

Moguće rješenje:

```
imena = [ "Ian", "Zack", "Lana", "Ivo", "Leo", "Iva", "Ana", "Mia" ]
uvuci = [ 0, 2, 4, 5, 4, 6, 2, 0 ]

for i in range(len(imena)):
    for j in range(uvuci[i]):
        print(" ", end="")

    print(imena[i])
```