

Python – vježbe 5

24. maj 2021.

Liste

1. Napisati program koji od korisnika traži unos liste cijelih brojeva. Nakon toga, program radi sljedeće stavke:
 - (a) Štampa broj elemenata liste
 - (b) Štampa posljednji element liste.
 - (c) Štampa listu sa obrnutim redosljedom elemenata.
 - (d) Štampa da li lista sadrži broj 13 ili ne sadrži.
 - (e) Štampa koliko ima elemenata 10 u listi.
 - (f) Briše posljednji element liste, a preostale podatke sortira a zatim ih štampa.
 - (g) Štampa koliko ima cijelih brojeva u listi koji su manji od 11.
2. Napisati program koji generiše listu od 20 slučajnih brojeva između 1 i 100, nakon čega:
 - (a) Štampa listu.
 - (b) Štampa srednju vrijednost elemenata liste.
 - (c) Štampa najmanji i najveći element liste.
 - (d) Štampa drugi najmanji i drugi najveći element liste.
 - (e) Štampa koliko ima neparnih cijelih brojeva u listi.
3. Posmatra se lista od tri elementa: `[4, 11, 15]`. Potrebno je odraditi sljedeće manipulacije:
 - (a) Postaviti drugi element na 22.
 - (b) Dodati 55, 67 i 82 na kraj liste.
 - (c) Obrisati prvi element liste.
 - (d) Sortirati listu.
 - (e) Duplirati listu.
 - (f) Ubaciti 25 na mjesto trećeg elementa liste.
4. Napisati program koji traži od korisnika da unese listu sa elementima između 1 i 23, od ukupno 4 elementa. Nakon toga je potrebno da se svi brojevi veći od 15 zamijene sa 0.
5. Napisati program koji od korisnika traži unos liste stringova, a koji zatim kreira listu koja sadrži iste stringove, ali bez prvog slova.
6. Korišćenjem petlji, kreirati liste:
 - (a) Listu brojeva od 10 do 100.
 - (b) Listu koja sadrži kvadrate brojeva od 20 do 50.

(c) Listu koja sadrži ponovljena slova u zavisnosti od njihove pozicije u engleskom alfabetu: ['a', 'bb', 'ccc', ...]. Posljednji element je string sastavljen od 26 slova z.

7. Napisati program koji traži unos dvije liste, L i M, sastavljene od cjelobrojnih elemenata (podrazumijevano iste dužine). Program kreira listu čiji su elementi sume odgovarajućih elemenata iz lista L i M. Na primjer, za $L=[5, 6, 7]$ i $M=[10, 20, 30]$, generiše se nova lista $N=[15, 26, 37]$.
8. Napisati program koji za unijeti cio broj kreira listu sastavljenu od njegovih prostih činilaca.
9. Pri bacanju dvije kockice, vjerovatnoće pojavljivanja brojeva dobijenih sabiranjem brojeva sa kocki nijesu iste. Na primjer, broj 12 se može dobiti sa vjerovatnoćom od 3%, a broj 7 sa vjerovatnoćom o 17%. Napišite program koji simulira bacanje dvije kocke i koji određuje vjerovatnoću pojavljivanja ishoda 2, 3, 4, ..., 12.
10. Napišite program koji rotira elemente liste tako da se element na prvom indeksu pomjera na drugi, drugi na treći, itd., sve do posljednjeg koji se pomjera na prvi indeks.
11. Korišćenjem for petlje kreirajte listu prikazanu ispod, koja se sastoji od jedinica razdvojenih rastućim brojem nula. Posljednje dve jedinice treba da budu razdvojene sa deset nula:
[1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, ...]
12. Napišite program koji generiše 100 slučajnih brojeva koji su ili 0 ili 1. Zatim odredite najduži uzastopni niz nula. Na primjer, najduži uzastopni niz nula u listi [1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0] je 4.
13. Napišite program koji briše višestruka pojavljivanja podataka u listi, tako da se podaci pojavljuju samo jednom. Na primjer lista [1, 1, 2, 3, 4, 3, 0, 0] će poslije takvog brisanja postati [1, 2, 3, 4, 0].
14. (D) Napišite program koji traži od korisnika da unese dužinu u stopama. Program treba da korisniku pruži mogućnost da bira jednicu u koju će ta dužina biti konvertovana. Jedinice mogu biti inči, jarde, milje, milimetri, centimetri, metri ili kilometri. Iako ovo može biti urađeno if naredbama, mnogo je kraće ako koristimo liste i for petlje, a takođe je mnogo lakše ako želimo da dodamo nove jedinice za konverziju.
15. Postoji neprobojna šifra pod nazivom „jednokratni ključ“. Način na koji ona radi je da pomjerate svako slovo poruke za slučajaj broj između 1 i 26, sa preklapanjem azbuke ako je potrebno. Na primjer ako je slovo koje treba šifrirati y a slučajni pomak 5, tada je novo slovo d. Svako slovo dobija svoj pomak, pa nam je potrebno onoliko slučajnih pomaka koliko ima slova u poruci. Kao primer recimo da je korisnik uneo riječ *secret*. Program treba da generiše 6 slučajnih brojeva između 1 i 26 (razmatračemo samo engleski alfabet). Pretpostavimo da su generisani brojevi 1, 3, 2, 10, 8, i 2. Šifrovana poruka će biti *thebmv*.
 - Napišite program koji traži od korisnika da unese poruku, pa tu poruku šifrira korišćenjem gornjeg metoda. Najprije konverujte string u mala slova. Svaki *space* znak i znak interpunkcije treba da ostane nepromijenjen. Na primjer *Secret!!!* postaje *thebmv!!!* korišćenjem gornjih pomaka.
 - Napišite program koji dešifruje poruke šifrovane na gornji način.

Razlog zašto se ovaj metod zove „jednokratni ključ“ je zato što se slučajni pomaci koriste samo jedanput. Šifra bi mogla biti lako razbijena ako bi se isti pomaci koristili u više poruka. Štaviše, ovaj metod je potpuno siguran samo ako su slučajni brojevi stvarno slučajni, što nije slučaj sa *randint* funkcijom koja daje takozvane pseudo slučajne brojeve. Za potrebe kriptografije postoje drugi pouzdaniji načini generisanja slučajnih brojeva.

Liste – predlog rješenja

1.

```
L = eval(input('Unesite listu: '))

print(' (a) Broj elemenata liste je: ', len(L))
print(' (b) Poslednji element je:', L[-1])
print(' (c) Obrnuta lista je:', L[::-1] )

ind = 0
for e in L:
    if e == 13:
        ind = 1

if ind == 1:
    print(' (d) Ima broja 13 u listi')
else:
    print(' (d) Nema broja 13 u listi')

print(' (e) Brojeva 10 ima ukupno: ',L.count(10))

L1 = L # cuvamo za poslednju stavku
del L[-1]
L.sort()
print(' (f) Sortirana lista je: ', L)

BR = 0
for e in L1:
    if e < 11:
        BR += 1
print(' (g) Brojeva manjih od 11 je: ',BR)
```

2. `from random import randint`

```
L = []
for i in range(20):
    L.append(randint(1,100))

print(' (a) lista je:',L)
print(' (b) srednja vrijednost elemenata je:',sum(L)/len(L))
print(' (c) najmanji element je: ', min(L), 'a maksimalni je:', max(L))

L1 = L
L1.sort()
print(' (d)', 'Drugi najveći je: ', L1[-2], 'a drugi najmanji je', L1[1])

BRnep = 0
for e in L:
    if e%2 == 1:
        BRnep += 1
print(' (e) Neparnih elemenata je: ', BRnep)
```

3. `L = [4, 11, 15]`

```
#(a)
L[1] = 22
```

```

#(b)
L = L + [55, 67, 82]
# L = L + [55] + [67] + [82]
# moze i:
# L.append(55)
# L.append(67)
# L.append(82)

```

```

#(c)
del L[0]

```

```

#(d)
L.sort()

```

```

#(e)
L = L*2

```

```

#(f)
L.insert(2,25)

```

4. `L = eval(input('Unesite listu od 4 broja izmedju 1 i 23: '))`

```

for i in range(len(L)):
    if L[i] > 15:
        L[i] = 0

```

```

print('Lista je: ', L)

```

5. `L = eval(input('Unesite listu stringova: ')) #!!!`

```

L1 = []

```

```

for s in L:
    L1.append(s[1:])

```

```

print(L1)

```

6. `La = []`

```

Lb = []

```

```

Lc = []

```

```

for n in range(10,101):
    La.append(n)

```

```

for n in range(20,51):
    Lb.append(n**2)

```

```

alfabet = 'abcdefghijklmnopqrstuvwxyz'

```

```

for i in range(len(alfabet)):
    Lc.append(alfabet[i]*(i+1))

```

```

print(La)

```

```

print(Lb)

```

```

print(Lc)

```

7. `L = eval(input('Unesite listu L: '))`

```

M = eval(input('Unesite listu M iste dužine: '))

```

```

N = []

```

```
for i in range(len(L)):
    N.append(L[i]+M[i])
```

```
print(N)
```

8. `N = eval(input('Unesite cio broj: '))`

```
L = []
```

```
prosti_cinilac = 2
```

```
while N != 1:
```

```
    if N%prosti_cinilac == 0:
```

```
        L.append(prosti_cinilac)
```

```
        N = (N / prosti_cinilac)
```

```
    else:
```

```
        prosti_cinilac = prosti_cinilac + 1
```

```
print(L)
```

9. `from random import randint`

```
kocka1 = []
```

```
kocka2 = []
```

```
zbirovi = []
```

```
BR_EKSP = 100000
```

```
for i in range(BR_EKSP):
```

```
    kocka1.append(randint(1,6))
```

```
    kocka2.append(randint(1,6))
```

```
for i in range(BR_EKSP):
```

```
    zbirovi.append(kocka1[i]+kocka2[i])
```

```
    #zbir rezultata na kockama u i-tom bacanju
```

```
#print(zbirovi)
```

```
ishodi = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

```
vjer_ishoda = []
```

```
for i in ishodi:
```

```
    vjer_ishoda.append(zbirovi.count(i)*100/BR_EKSP)
```

```
print('Ishodi', ishodi)
```

```
print('Vjerovatnoce', vjer_ishoda)
```

10. `pom = L[-1]`

```
del L[-1]
```

```
L.insert(0,pom)
```

```
print(L)
```

11. `L = [1]`

```
for br_nula in range(0,11):
```

```
    for i in range(br_nula):
```

```
        L.append(0)
```

```
        L.append(1)
```

```
print(L)
```

12. `from random import randint`

```
L=[randint(0,1) for i in range(100)]

LB =[]

i = 0
while i < len(L):
    if L[i] == 0:
        j = i
        br = 0
        while j < len(L):
            br = br + 1
            if L[j] != 0:
                break
            j = j + 1
        LB.append(br)
    i = i +1
print(L)
print('Trazeni rezultat je:', max(LB))
```

13. `L = [1, 1, 2, 3, 4, 3, 0, 0]`

```
L1 = []

for i in range(len(L)):
    ind = 0
    for j in range(len(L1)):
        if L[i]==L1[j]:
            ind = 1
    if ind == 0:
        L1.append(L[i])

print(L1)
```

1 Liste – malo naprednije teme

1. Napišite program koji traži od korisnika da unese neki engleski tekst, a zatim prebrojava koliko ima članova u tom tekstu. Članovi su riječi 'a', 'an', i 'the'.
2. Napišite program koji omogućava korisniku da unese pet brojeva (pročitanih kao string). Kreirajte string koji se sastoji od unijetih brojeva sa „+“ znakom kao separatorom. Na primer, ako korisnik unese 2, 5, 11, 33, i 55, onda će string biti '2+5+11+33+55'.
3. Napisati program koji traži od korisnika da unese jednu rečenicu a zatim štampa treću riječ iz te rečenice.
4. Napisati program koji traži od korisnika da unese jednu rečenicu a zatim štampa *svaku treću* riječ iz te rečenice.
5. Napisati program koji traži od korisnika da unese jednu rečenicu a zatim na slučajan način razmješta riječi te rečenice. Nije potrebno voditi računa o znacima interpunkcije i velikom slovu na početku rečenice.
6. (D) Uraditi prethodni zadatak, ali sada voditi računa da rečenica počinje velikim slovom, te da originalna prva riječ ne treba da počne velikim slovom ako se nađe negdje u sredini rečenice, a takođe i tačka treba da bude na pravom mjestu.
7. Napišite jednostavan program za „Objavu dana“. Program treba da sadrži listu objava, pa kada se izvrši štampa jednu slučajno izabranu.
8. Napišite program za izvlačenje brojeva u igri LOTO. Izvlačenje se sastoji od šest slučajno izabranih brojeva između 1 i 48.
9. Napišite program kojim se procenjuje prosječan broj izvlačenja potrebnih da korisnički tiket od šest brojeva ostvari dobitak u igri loto u kojoj se biraju brojevi od 1 do 10. Da to uradite izvršite 1000 puta petlju u kojoj se generiše korisnički tiket i simulira izvlačenje sve dok korisnički tiket nije izvučen. Izračunajte prosečan broj izvlačenja koja su bila potrebna u 1000 izvršenih simulacija.
10. Napišite program kojim se simulira izvlačenje imena iz šešira. U ovom izvlačenju svaka osoba može imati više pojavljivanja u šeširu. Omogućite korisniku da unese listu imena i listu brojeva pojavljivanja imena u šeširu, a zatim štampajte pobjednika.
11. (D) Napišite jednostavnu kviz igru koja ima listu od 10 pitanja i listu odgovora na ta pitanja. U igri korisnik dobija 4 slučajno odabrana pitanja. Program postavlja pitanja jedno po jedno i odmah saopštava korisniku da li je odgovor tačan ili pogrešan. Na kraju program štampa koliko je bilo tačnih odgovora.
12. Napišite program za cenzurisanje teksta. Omogućite korisniku da unese neki tekst, a zatim da program štampa tekst sa zvjezdicama umjesto slova u zabranjenim riječima. Za testiranje programa koristite sljedeće reči kao one koje treba cenzurisati: *darn, dang, freakin, heck, i shoot*, kao što je prikazano u primjeru:

```
Enter some text: Oh shoot, I thought I had the dang problem
figured out. Darn it. Oh well, it was a heck of a freakin
try.

Oh *****, I thought I had the **** problem figured out.
**** it. Oh well, it was a **** of a ***** try.
```
13. Napišite program koji koristi `shuffle` za kreiranje slučajnog anagrama učitano stringa.
14. Napišite program koji preuzima od korisnika string koji je potencijalno telefonski broj. Program treba da štampa *Ispravan* ako odluči da je string predstavlja stvaran telefonski broj, a *Neispravan* ako string

ne odgovara stvarnom telefonskom broju. Telefonski broj će se smatrati ispravnim ako ima oblik *abc-def-hijk* ili *1-abc-def-hijk*. Crtice moraju biti uključene, telefonski broj može sadržati samo cifre i crtice, a broj cifara u svakoj grupi mora biti korektan. Testirajte program sa sljedećim podacima:

Unesite telefonski broj: 1-301-447-582

Ispravan

Unesite telefonski broj: 301-447-5820

Ispravan

Unesite telefonski broj: 301-4477-5820

Neispravan

Unesite telefonski broj: 3X1-447-5820

Neispravan

Unesite telefonski broj 3014475820

Neispravan

15. Neka je L lista stringova. Napišite naredbu za sastavljanje liste (list-comprehensions) koja kreira novu listu od liste L na sljedeće načine:
 - Listu koja sadrži stringove s iz liste L sa uklonjenim prvim slovom
 - Listu koja sadrži dužine stringova iz L
 - Listu koja se sastoji od onih stringova iz L koji su dužine najmanje 3 znaka
16. Iskoristite list-comprehensions da napravite listu koja se sastoji od svih palindromskih brojeva između 100 i 1000.
17. Iskoristite list-comprehensions da napravite donju listu, koja se sastoji od jedinica razdvojenih rastućim brojem nula. Posljednje dvije jedinice treba da budu razdvojene sa deset nula:
[1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1,]
18. Neka je L=[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]. Korišćenjem list-comprehension metode napravite listu razlika među uzastopnim brojevima iz L. Zatim nađite maksimalnu razliku kao i procenat slučajeva kada je razlika 2.
19. Napišite program koji nalazi srednju vrijednost svih elemenata u nekoj 4×4 listi cijelih brojeva.
20. Napišite program koji kreira listu od 10×10 slučajnih brojeva između 1 i 100. Zatim je potrebno:
 - Štampajte listu.
 - Nađite najveću vrijednost u trećem redu.
 - Nađite najmanju vrijednost u šestoj koloni.
21. Napišite program koji kreira i štampa listu dimenzija 8×8 čiji su elementi naizmjenično 1 i 2 , počevši od 1 u gornjem lijevom uglu.
22. Napišite program koji provjerava da li je lista dimenzija 4×4 magičan kvadrat. Kod magičnog kvadrata svaki red, svaka kolona kao i dve dijagonale imaju istu vrijednost sume svojih elemenata.
23. Sljedeći program je koristan kao dio igre „Podmornice“. Pretpostavimo da imate listu dimenzija 5×5 koja sadrži nule i jedinice. Tražite od korisnika da unese red i kolonu. Ako je podatak u tom redu i koloni liste jednak jedinici, program treba da štampa *Pogodak*, u suprotnom štampa *Promašaj*.
24. Ovaj zadatak može da posluži za igru „Memorija“. Slučajno generišite listu dimenzija 6×6 koja sadrži tekstualne znakove tako da se svaki pojavljuje tačno dva puta. Jedan primjer je prikazan ispod:


```

@ 5 # A A !
5 0 b @ $ z
$ N x ! N z
0 - + # b :
- : + c c x

```

25. Ovaj zadatak može biti koristan za implementaciju igrača u raznim kompjuterskim igrama. Napišite program koji kreira jednu 5×5 listu koja se sastoji od nula i jedinica. Vaš program treba da slučajno izabere jednu od pozicija u listi koja sadrži nulu i da je promijeni u jedinicu. Ako su svi podaci u listi jedinica, program treba to da sopšti. *[Pomoć: jedan način da se ovo uradi je da se kreira nova lista čiji će elementi biti koordinate svih jedinica u listi, pa korišćenjem choice metoda slučajno bira jedna od njih. Možete koristiti listu sa dva elementa da predstavite koordinate.]*

26. Postoji samo jedan petocifreni broj n takav da svaki od sljedećih brojeva ima samo jednu cifru koja je sa njim zajednička na istoj poziciji. Pronađite broj n :

01265, 12171, 23257, 34548, 45970, 56236, 67324, 78084, 89872, 99414

27. Obično se elementi dvodimenzione liste adresiraju pomoću redova i kolona, kao što je ispod prikazano (lijevo). Drugi način bi bio okorišćenje adresiranja kao što je prikazano desno:

(0, 0)	(0, 1)	(0, 2)	0	1	2
(1, 0)	(1, 1)	(1, 2)	3	4	5
(2, 0)	(2, 1)	(2, 2)	6	7	8

- Napišite kod koji prevodi iz lijeve u desnu prezentaciju. Operacije // i % su korisne za ovu transformaciju.
- Provjerite da li vaš kod radi za bilo koje dimenzije liste. Napišite kod koji prevodi iz desne u levu prezentaciju.

2 Liste – naprednije teme – predlog rješenja

```
1. s = input('Unesite tekst: ')

s=s.lower(a
listaRijeci = s.split()

brojClanova = listaRijeci.count('a')+\
               listaRijeci.count('the') +\
               listaRijeci.count('an')
print(brojClanova)
```

```
2. s = input('Unesite tekst: ')

listaBrojeva = s.split(',')
s1 = '+'.join(listaBrojeva)

print(s1)
```

```
3. from string import punctuation
s = input('Unesite tekst: ')

for c in punctuation:
    s = s.replace(c, '')

L = s.split()
print(L[2])
```

```
4. from string import punctuation
s = input('Unesite tekst: ')

for c in punctuation:
    s = s.replace(c, '')

L = s.split()

for i in range(0, len(L), 3):
    print(L[i])
```

Za vježbu uraditi sa `while` petljom.

```
5. from string import punctuation
from random import shuffle

s = input('Unesite tekst: ')
s = s.lower()

for c in punctuation:
    s = s.replace(c, '')

L = s.split()
shuffle(L)
s1=' '.join(L)

print()
print(s1)
```

6. Za vježbu.

7. `from random import shuffle`

```
L = ['Objava 1', 'Novosti na ETF-u',  
     'Python takmicenje']
```

```
shuffle(L)
```

```
print(L[0])
```

8. `# prvi nacin`

```
from random import sample
```

```
L = [i for i in range(1,49)]
```

```
print(sample(L,6))
```

```
# drugi nacin
```

```
from random import shuffle
```

```
L = [i for i in range(1,49)]
```

```
shuffle(L)
```

```
print(L[0:6])
```

9. `# prvi nacin`

```
from random import sample
```

```
L = [i for i in range(1,11)]
```

```
dobitnaKomb = sample(L,6)
```

```
dobitnaKomb.sort()
```

```
listaIzvlacenja = []
```

```
for ee in range(1000):
```

```
    brojIzvlacenja = 0
```

```
    for i in range(1000):
```

```
        tiket = sample(L,6)
```

```
        tiket.sort()
```

```
        if tiket == dobitnaKomb:
```

```
            brojIzvlacenja += 1
```

```
    listaIzvlacenja.append(brojIzvlacenja)
```

```
print('prosjecan br. izvl.: ',
```

```
      sum(listaIzvlacenja)/len(listaIzvlacenja))
```

Za vježbu, modifikovati unutrašnju petlju u `while`.

10. Za vježbu.

11. Za vježbu.

12. Napišimo jedno nepotpuno rješenje:

```
#s = input('Unesite tekst: ')
```

```
s = 'Oh shoot, I thought I had the dang \  
     problem figured out. Darn it. \  
     '
```

```
Oh well, it was a heck of a freakin try.'
```

```
listaC = ['darn', 'dang', 'freakin', 'heck', 'shoot']

listaRijeci = s.split()

for tekst in listaC:
    for i in range(len(listaRijeci)):
        if listaRijeci[i].lower() == tekst.lower():
            listaRijeci[i] = '*'*len(listaRijeci[i])

cenzurisano = ' '.join(listaRijeci)
print(cenzurisano)
```

Kako da riješimo problem sa znakovima interpunkcije?

```
#s = input('Unesite tekst: ')

s = '''Oh shoot, I thought I had the dang problem
figured out. Darn it. Oh well, it was a heck of a
freakin try.'''

listaC = ['darn', 'dang', 'freakin', 'heck', 'shoot']

for tekst in listaC:
    s = s.replace(tekst, '*'*len(tekst))
    s = s.replace(tekst.capitalize(), '*'*len(tekst))

print(s)

# za vježbu, odradite sa for petljom po stringu
# uskoro cemo nauciti kako da sami napravimo funkciju
```

13. `from random import shuffle`

```
s = 'Nekitekst'
L = list(s)
shuffle(L)

NoviString = ' '.join(L)
print(NoviString)
```

14. Za vježbu

15.

```
L = ['Jedan', 'Dva', 'Tri', 'Cetiri']

L1 = [s[1:] for s in L]
L2 = [len(s) for s in L]
L3 = [s for s in L if len(s) == 3]

# stampanje rezultata
print(L1)
print(L2)
print(L3)
```

```

16. L = [broj for broj in range(100, 1001) if str(broj) == str(broj)[::-1]]
    print(L)

17. L = list(''.join(['1'+ '0'*BrojNula for BrojNula in range(0,11)]+'1'))
    print(L)

18. L = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

    L1 = [L[i+1] - L[i] for i in range(len(L)-1)]

    MaxRazlika = max(L1)
    BrojDvojki = L1.count(2)
    ProcenatDvojki = 100 * BrojDvojki/len(L1)

    print(L1)
    print(MaxRazlika)
    print(ProcenatDvojki)

19. L = [[2, 3, 5, 7], [11, 13, 17, 19], [23, 29, 31, 37], [41, 43, 47, 50]]

    Suma = 0

    for i in range(0, 4):
        for j in range(0, 4):
            Suma += L[i][j]

    print(Suma/16)

    # drugi nacin?

    Suma = sum([L[i][j] for i in range(0, 4) for j in range(0, 4)])
    print(Suma/16)

20. Za vježbu

21. s = '12'

    L = []
    for i in range(8):
        L.append(list(s*4))

    print(L)

22.

L = [[16, 2, 3, 13],
      [5, 11, 10, 8],
      [9, 7, 6, 12],
      [4, 14, 15, 1]]

ind = 0
SumaVrsta = sum(L[0])

# provjera po vrstama:
for i in range(len(L)):
    if SumaVrsta != sum(L[i]):
        ind = 1

```

```

# provjera po kolonama:
for j in range(len(L)):
    SumaKolona = 0
    for i in range(len(L)):
        SumaKolona += L[i][j]
    if SumaVrsta != SumaKolona:
        ind = 1

# provjera po glavnoj dijagonali
# (moglo je u gornjoj petlji)

SumaDijagonala = 0

for i in range(len(L)):
    SumaDijagonala += L[i][i]

if SumaVrsta != SumaDijagonala:
    ind = 1

# provjera po sporednoj dijagonali
# moglo se spojiti sa gornjim petljama

SporednaDijagonala = 0
for i in range(len(L)):
    SporednaDijagonala += L[i][len(L)-1-i] #*mozda tesko

if SporednaDijagonala != SumaVrsta:
    ind = 1

if ind == 0:
    print('Jeste magican!')
else:
    print('Nije magican!')

```