LISTA i MATRICA (lista lista), RJEČNIK

#kreiranje prazne liste

lista=[]

#dodavanje stavki u listu

lista.append("Split")

lista.append("Solin")

#pristup preko indeksa

print(lista[1])

#ispis svih elemenata preko indeksa

for idx in range(0,len(lista)):

print(lista[idx])

#ispis svih elemanata bez indeksa

for stavka in lista:

print (stavka)

#matrica

lista1=[1,2,3]

lista2=[3,4,5]

matrica=[]

matrica.append(lista1)

matrica.append(lista2)

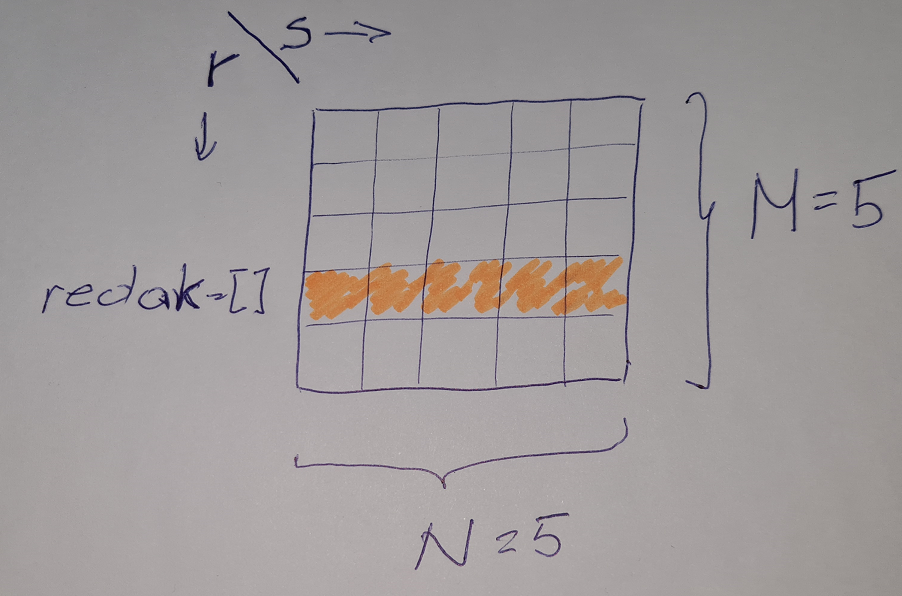
print(matrica)

matr=[[1,2,3],[3,4,5]]

print(matr)

#ispis određenog elementa

print(matr[1][2])



#rječnik

rjecnik={}

rjecnik[3]="Split"

rjecnik[5]=[1,2,3,4]

print (rjecnik)

1. Napisati funkciju koja prima listu brojeva. Funkcija vraća listu koja sadrži svaki broj samo jednom. Npr. za listu [6,5,6,4,8,4], funkcija vraća [6,5,4,8].

def lista\_unique(lista):

nova\_lista=[]

for broj in lista:

if broj not in nova\_lista:

nova\_lista.append(broj)

return nova\_lista

print(lista\_unique([6,5,6,4,8,4]))

1. Napisati program koji od korisnika čita string koji predstavlja tri cijela broja odvojena razmakom. Ako je string ispravnog formata (npr. "12 45 16"), program ispisuje zbroj brojeva. U suprotnom program ispisuje poruku grešci.

tekst="3 62 -12"

lista=tekst.split(" ")

if len(lista)!=3:

print("Greska!")

else:

print(int(lista[0])+int(lista[1])+int(lista[2]))

1. Napisati funkciju koja prima rječnik koji kao ključeve ima brojeve, a kao vrijednosti liste brojeva. Funkcija vraća True ako je svaki ključ jednak zbroju pridružene liste.

rj={2:[1,1], 4:[1,2,1], 6:[2,2,2]}

def Test (rj):

for k in rj:

suma=0

for broj in rj[k]:

suma+=broj

if k!=suma:

return False

return True

print (Test(rj))

1. Napisati funkciju koja prima listu brojeva i listu stringova. Funkcija ispisuje samo stringove na indeksima zadanim u listi brojeva.

def oper\_string(brojevi, stringovi):

for i in brojevi:

print(stringovi[i])

oper\_string([1,3,4],["tekst","Vlak","test","u","snijegu","hh"])

1. Napisati funkciju koja prima listu brojeva i dva broja M i N koji predstavljaju broj redaka i dužinu retka. Funkcija vraća novu matricu (veličine MxN određene brojem redaka i dužinom retka) koja sadrži sve brojeve iz liste i ostatak popunjen nulama.

def popunjena\_mat(lista,M,N):

matrica=[]

i=0 #indeks zadnje očitanog elementa u listi

for r in range (0,M):

#kreiranje praznog retka

redak=[]

#popunjavanje stupaca

for s in range (0,N):

#provjera ima li još elemenata u listi

if i<len(lista):

#dodaj element iz liste

redak.append(lista[i])

#povecaj indeks u listi na sljedeci

i+=1

#nema više elemenata u listi pa dodaj 0

else:

redak.append(0)

#dodavanje retka u listu

matrica.append(redak)

return matrica

print(popunjena\_mat([1,2,3,4,5],3,3))

print(popunjena\_mat([1,2,3,4,5,6,7,8,9,10,11],3,3))

matrica=popunjena\_mat([1,2,3,4,5,6,7,8,9,8,7],4,4)

for redak in matrica:

print (redak)

1. Napisati funkciju koja prima listu brojeva i listu booleana jednake dužine. Lista booleana ima točno dva elementa postavljena na True. Ta dva elementa označavaju početak i kraj liste brojeva koju funkcija treba „izbaciti“ iz originalne liste (funkcija vraća novu listu). Na primjer, za liste [ 6, 2, 8, 4, 9 ] i [ F, T, F, T, F ], funkcija vraća listu [ 6, 9 ].

def dio\_liste(brojevi, booleani):

nova\_lista=[]

puni=True

for i in range(0,len(booleani)):

if booleani[i]==True:

puni=not puni

#continue preskače ostatak i nastavlja sljedeću iteraciju

continue

if puni==True:

nova\_lista.append(brojevi[i])

return nova\_lista

print(dio\_liste([6,2,8,4,9],[False,True,False,True,False]))

1. Napisati funkciju koja prima dva stringa sastavljena od cijelih pozitivnih brojeva odvojenih zarezom. Oba stringa sadrže isti broj brojeva. Funkcija vraća string (istog formata) gdje su elementi zbroj dva broja iz prvog i drugog stringa. Npr. za „3,14,2,10“ i „6,11,5,5“, funkcija vraća „9,25,7,15“.

def concat(s1,s2):

s1\_list=s1.split(',')

s2\_list=s2.split(',')

s3=""

for p in range(0,len(s1\_list)):

if p>0: s3=s3+","

suma=int(s1\_list[p])+int(s2\_list[p])

s3=s3+str(suma)

return s3

s1="3,14,2,10"

s2="6,11,5,5"

print(concat(s1,s2))

1. Napisati funkciju koja prima rječnik koji kao ključeve ima brojeve, a kao vrijednosti liste brojeva. Funkcija vraća True ako je svaki ključ jednak prosjeku brojeva u pridruženoj listi.

def zbroj(rjecnik):

for kljuc in rjecnik:

zbroj=0

for broj in rjecnik[kljuc]:

zbroj+=broj

if kljuc!=zbroj/len(rjecnik[kljuc]):

return False

return True

rjecnik={2:[1,2,3],4:[2,4,6],3:[1,8,0]}

print(zbroj(rjecnik))

1. Napisati funkciju koja prima jedan string i dva broja A i B. Funkcija vraća True ako string sadrži A samoglasnika i B suglasnika. Npr. za string „ab cd e“, A=2 i B=3, funkcija vraća True.

tekst='ab cd e'

def Test(tekst,A,B):

br\_sam=0

br\_sug=0

sam=['A','E','I','O','U']

for znak in tekst:

if znak.isalpha():

if znak.upper() in sam:

br\_sam+=1

else:

br\_sug+=1

if A==br\_sam and B==br\_sug:

return True

return False

print(Test(tekst,2,3))

1. Napisati funkciju koja prima listu brojeva. Funkcija vraća listu brojeva iz primljene liste koji su djeljivi sa svim prethodnim brojevima iz primljene liste. Npr. za listu [ 2, 3, 6, 12, 15 ], funkcija vraća listu [6, 12].

def djelj\_broj(lista):

nova\_lista=[]

#ide od 1 jer prvi element nema prethodnika

for i in range(1,len(lista)):

#pretpostavka da je djeljiv sa svima prije

djeljiv=True

#provjera je li djeljiv sa svim prethodnicima

for j in range(0,i):

#ako nije djeljiv samo sa jednim stavi False i prekini dalje

if lista[i]%lista[j]!=0:

djeljiv=False

break

#dodaj ga listu samo ako je ostalo True

if djeljiv==True:

nova\_lista.append(lista[i])

return nova\_lista

print(djelj\_broj([2,3,6,12,15]))

1. Napisati funkciju koja prima paran broj N i vraća matricu veličine NxN ispunjenu sa jedinicama i nulama. Gornja polovica matrice je ispunjena jedinicama.

def popunjena\_mat(N):

matr=[]

for r in range(0, N):

redak=[]

for s in range(0,N):

if r<N/2:

redak.append(1)

else:

redak.append(0)

matr.append(redak)

return matr

#print(popunjena\_mat(6))

mat=popunjena\_mat(6)

for redak in mat:

print(redak)

1. Napisati funkciju koja prima matricu (liste lista) i vraća True ako je prva matrica jednaka transponiranoj drugoj matrici (retci i kolone su joj zamijenjene).

def transp\_mat(mat1,mat2):

for r in range(0,len(mat1)):

for s in range(0,len(mat1)):

if mat1[r][s]!=mat2[s][r]:

return False

return True

print(transp\_mat([[1,2,3],[4,5,6],[7,8,9]],[[1,4,7],[2,5,8],[3,6,9]]))

print(transp\_mat([[1,2,3],[4,5,6],[7,8,9]],[[1,4,7],[2,5,8],[3,6,1]]))

1. Napisati funkciju koja prima matricu (listu lista) i vraća listu sastavljenu od svih brojeva u matrici.

def mat\_to\_lista(mat):

lista=[]

for red in mat:

for broj in red:

lista.append(broj)

return lista

#duže rješenje

def mat\_to\_lista\_2(mat):

lista=[]

#r čita retke

for r in range(0,len(mat)):

#s čita stupce

for s in range(0,len(mat[r])):

lista.append(mat[r][s])

return lista

print(mat\_to\_lista([[1,3,4],[],[-1,-2]]))

print(mat\_to\_lista\_2([[1,3,4],[],[-1,-2]]))

1. Napisati funkciju koja prima dvije matrice (liste lista) i vraća True ako su obje matrice jednake po sadržaju i dimenzijama.

def transp\_mat(mat1,mat2):

if len(mat1)!=len(mat2):

return False

for r in range(0,len(mat1)):

if len(mat1[r])!=len(mat2[r]):

return False

for s in range(0,len(mat1)):

if mat1[r][s]!=mat2[r][s]:

return False

return True

print(transp\_mat([[1,2,3],[4,5,6],[7,8,9]],[[1,2,3],[4,5,6],[7,8,9]]))

print(transp\_mat([[1,2,3],[4,5,6],[7,8,9]],[[1,2,3],[4,5,6],[7,8]]))

print(transp\_mat([[1,2,3],[4,5,6],[7,8,9]],[[4,5,6],[7,8,9]]))

1. Napisati funkciju koja prima dvije liste brojeva jednake dužine. Funkcija vraća novu listu sastavljenu od brojeva koji se uzimaju iz primljenih lista. Funkcija za svaki element u novoj listi bira manji od dva elementa u pripremljenim listama na istom indeksu.

l1=[1,2,3,4,5,6]

l2=[2,1,4,5,6,1]

def Test(l1,l2):

l=[]

for i in range (len(l1)):

if l1[i]<l2[i]:

l.append(l1[i])

else:

l.append(l2[i])

return l

print(Test(l1,l2))

1. Napisati funkciju koja prima rječnik sa stringovima kao ključevima i listama brojeva kao vrijednostima. Za svaki ključ u rječniku funkcija računa najveći broj iz pridružene liste. Funkcija vrača rječnik sastavljen od istih ključeva i pridruženih najvećih brojeva.

rj={'l1':[1,2,3], 'l2':[6,7,8], 'l3':[11,12]}

def Test(rj):

novi={}

for k in rj:

lista=rj[k]

maks=lista[0]

for i in range(1,len(lista)):

if lista[i]>maks:

maks=lista[i]

novi[k]=maks

return novi

print(Test(rj))

1. Napisati funkciju koja prima string koji predstavlja cijele brojeve odvojene razmakom. Brojevi mogu biti negativni (imati ’-’ ispred sebe). Funkcija vraća koliko brojeva je zapisano u stringu.

Primjer stringa: "3 62 -12 8".

#pretpostavka da nema slova i znakova osim '-'

def brojevi\_u\_tekstu(tekst):

#string se rastavlja i sprema u listu

lista=tekst.split(" ")

print(lista)

return len(lista)

tekst="3 62 -12 8"

print(brojevi\_u\_tekstu(tekst))

1. Napisati funkciju koja prima jedan string. String može biti „X plus Y“ ili „X minus Y“, gdje su X i Y dva cijela broja (mogu imati više znamenki ili predznak). Funkcija vraća zbroj ili razliku brojeva X i Y.

def oper\_string(tekst):

lista=tekst.split(" ")

#print(lista)

if lista[1]=="plus":

return int(lista[0])+int(lista[2])

elif lista[1]=="minus":

return int(lista[0])-int(lista[2])

else:

return ("Pogrešan format.")

print(oper\_string("2 plus 3"))

print(oper\_string("231 minus 31"))

print(oper\_string("2 plu 3"))

1. Napisati funkciju koja prima string sastavljen od velikih slova. Funkcija vraća string gdje su slova zamijenjena znamenkama prema pravilima O=0, I=1 i E=3. Npr. za string "PREVELIKO", funkcija će vratiti "PR3V3L1K0". Nije dozvoljena upotreba gotovih funkcija poput replace().

def zamjena(tekst):

novi\_tekst=""

for slovo in tekst:

if slovo=="E":

novi\_tekst+="3"

elif slovo=="I":

novi\_tekst+="1"

elif slovo=="O":

novi\_tekst+="0"

else:

novi\_tekst+=slovo

return novi\_tekst

print(zamjena("PREVELIKO"))

1. Napisati funkciju koja prima string i vraća novi string koji nastaje tako da se iz primljenog stringa izbace svi zarezi.

def bez\_zareza(tekst):

lista=tekst.split(",")

novi\_tekst=""

for t in lista:

novi\_tekst+=t

return novi\_tekst

def bez\_zareza1(tekst):

novi\_tekst=""

for znak in tekst:

if znak != ",":

novi\_tekst+=znak

return novi\_tekst

print(bez\_zareza("ahhj67,ljj,kjk"))

print(bez\_zareza1("ahhj67,ljj,kjk"))

1. Napisati funkciju koja prima rječnik i jedan broj. Vrijednosti u rječniku su liste brojeva. Funkcija dodaje taj broj svim listama u rječniku.

def dodajN(rjecnik,N):

for kljuc in rjecnik:

rjecnik[kljuc].append(N)

return rjecnik

rjecnik={3:[1,2,3],9:[4,5,6,7,9],151:[100,151,150]}

print(dodajN(rjecnik,-3))

1. Napisati funkciju koja prima rječnik koji kao ključeve ima brojeve, a kao vrijednosti liste brojeva. Funkcija vraća True ako je svaki ključ jednak najvećem broju u pridruženoj listi.

def najveci(rjecnik):

for kljuc in rjecnik:

maksi=rjecnik[kljuc][0]

for broj in rjecnik[kljuc]:

if broj>maksi:

maksi=broj

if maksi!=kljuc:

return False

return True

rjecnik={4:[1,2,3],9:[4,5,6,7,8],152:[100,151,150]}

print(najveci(rjecnik))

1. Napisati funkciju koja prima rječnik i jedan broj. Vrijednosti u rječniku su liste brojeva. Funkcija vraća True ako se broj nalazi negdje u rječniku.

def imaliN(rjecnik,N):

for kljuc in rjecnik:

if N==kljuc or N in rjecnik[kljuc]:

return True

return False

rjecnik={3:[1,2,3],9:[4,5,6,7,9],151:[100,151,150]}

print(imaliN(rjecnik,100))

print(imaliN(rjecnik,107))

1. Napisati funkciju koja prima listu brojeva. Funkcija vraća rječnik koji će imati tri ključa: „parni“, „neparni“ i „nule“. Vrijednost svakog ključa je lista brojeva koji su parni, neparni ili nule.

brojke=[1,2,3,4,5,6,0,0,0]

def Test(brojke):

parni=[]

nepar=[]

nule=[]

rjk={}

for broj in brojke:

if broj==0:

nule.append(0)

elif broj%2==0:

parni.append(broj)

else:

nepar.append(broj)

rjk["parni"]=parni

rjk["neparni"]=nepar

rjk["nule"]=nule

return rjk

print(Test(brojke))

1. Napisati funkciju koja primljeni string rastavlja na niz stringova. Svaki element u nizu će biti dio originalnog stringa koji počinje velikim slovom i završava točkom. Npr. za string "Split.Zadar.Šibenik.računalo.Solin", funkcija vraća [ "Split", "Zadar",“Šibenik“ ].

def tekst\_veliko(tekst):

tekst\_lista=tekst.split(".")

lista=[]

for i in range(0,len(tekst\_lista)-1):

print(tekst\_lista[i])

if tekst\_lista[i][0].isupper():

lista.append(tekst\_lista[i])

return lista

tekst="Split.Zadar.Šibenik.računalo.Solin"

print(tekst\_veliko(tekst))

tekst="Split.Zadar.Šibenik.računalo.Solin."

print(tekst\_veliko(tekst))

tekst="Split.Zadar.Šibenik.računaloSolin"

print(tekst\_veliko(tekst))

1. Napisati funkciju koja prima listu lista brojeva. Funkcija od brojeva u listi konstruira rječnik koji kao ključeve ima zbroj svake pojedine liste, a svakom ključu je kao vrijednost pridružena ta lista.

def kreiraRjecnik(listalista):

rjecnik={}

for lista in listalista:

zbroj=0

for broj in lista:

zbroj+=broj

rjecnik[zbroj]=lista

return rjecnik

listalista=[[2,3,1],[5,5,7,1],[9,1,1,2]]

print(kreiraRjecnik(listalista))

#ne mogu biti dva ista kljuca

listalista=[[2,3,1],[3,2,1],[9,1,1,2]]

print(kreiraRjecnik(listalista))

1. Napisati funkciju koja prima string sastavljen od malih slova. Funkcija vraća string gdje su svi dvostruki suglasnici zamijenjeni jednim suglasnikom. Npr. za string „lookk“, funkcija će vratiti „look“.

def IzbacDuplSugl(tekst):

samogl=('a','e','i','o','u')

novitekst=tekst[0]

for i in range(1,len(tekst)):

if tekst[i] not in samogl:

if tekst[i]!=tekst[i-1]:

novitekst+=tekst[i]

else:

novitekst+=tekst[i]

return novitekst

print(IzbacDuplSugl("lookk"))

#Napisati funkciju koja primljeni string rastavlja na listu

#stringova. Svaki element u listi u orig. stringu započinje sa (

#( i završava sa ) zagradom. Pr. ab(cd)ef(g) vraća ["cd","g"]

tekst='ab(cd)ef(g)ee)(d'

def Test(tekst):

lista=[]

pamti=False

rijec=""

for znak in tekst:

if znak=='(':

pamti=True

rijec=""

elif znak==')':

if pamti==True:

lista.append(rijec)

pamti=False

else:

if pamti==True:

rijec+=znak

return lista