



Pretraživanje elemenata u listi

Lista unutar liste

- ▶ Do sada smo obrađivali jednodimenzionalne liste (brojevi ili riječi) npr. visine učenika, ali uz visinu učenika možemo dodati i njegovo ime, prezime, razred...
- ▶ Kod takvih podataka imat ćemo listu čiji elementi su liste
- ▶ Primjer: [[Marin, 175], [Maja, 159], [Dario, 182], [Dijana, 162]]
- ▶ Takvu listu možemo zamisliti kao tablicu (s četiri reda i dva stupca)

	Stupac 0	Stupac 1
Red 0	Marin	175
Red 1	Maja	159
Red 2	Dario	182
Red 3	Dijana	162

Problemi:

Pretraživanje:

Na kojoj poziciji je najviši učenik?

Sortiranje

Kako sortirati listu? (poredati od najmanjeg prema najvećem učeniku)

Zadatak 1.

Kreirajte listu dimenzija 4*3 (4 retka i 3 stupca) koja se sastoji od liste slučajnih brojeva od 1 do 10.

```
from random import*
L = []
for i in range(4):
    a = [randint(1, 10) for i in range(3)]
    L = L + [a]
print(L)
```

Zadatak 2.

Kreirajte listu dimenzija 3*3 (3 retka i 3 stupca) koja se sastoji od tri liste brojeva od 1 do 9.

a) Ispišite sve njezine elemente tako da izgledaju poput tablice.

1 2 3

4 5 6

7 8 9

b) Elemente na dijagonali tablice zamijenite s simbolom *(zvijezdica)

* 2 3

4 * 6

7 8 *

c) Sve elemente ispod dijagonale zamijenite znakom +

1 2 3

+ 5 6

+ + 9

Rješenje 2. zadatka

```
L = []
b = 1
for i in range(3): #kreiranje liste
    a = []
    for j in range(3):
        a.append(b)
        b += 1
    L = L + [a]
print(L)
```

```
for i in range(3):
    for j in range(3):
        print(L[i][j], end = " ")
    print()
```

Zadatak 3.

Kreirajte listu dimenzija 3×4 (3 retka i 4 stupca) koja se sastoji od tri liste slučajnih brojeva od 1 do 9.

a) Ispišite sve njezine elemente tako da izgledaju poput tablice.

4 2 7 9

1 5 2 8

3 3 7 6

b) Elemente u zadnjem stupcu tablice zamijenite s simbolom * (zvijezdica)

4 2 7 *

1 5 2 *

3 3 7 *

Zadatak 4.

Napišite program koji kreira listu u kojoj su spremljene visine od n učenika u razredu, a zatim unosi visinu x i provjerava nalazi li se visina x među unesenim visinama. Program će vratiti `True` ako se x nalazi u listi i `False` ako se ne nalazi zadana visina u listi.

Zadatak 5.

Napišite program koji kreira listu od n elemenata u kojoj su spremljene liste koje se sastoje od imena učenika i njihovih visina, a zatim unosi visinu x i provjerava koji učenici imaju visinu x . (Izbacit će novu listu s imenima traženih učenika, ako je lista prazna izbacit će poruku da nema takvih učenika)

Linearno pretraživanje

- **najjednostavnije i najčešće korišteno pretraživanje**
- **najsporije**
- **prolazi kroz sve elemente liste od prvog do posljednjeg i traženje zadanog elementa**
- **ako imamo puno elemenata pretraga će trajati dosta dugo.**

Zadatak 6.

Napišite program koji kreira listu u kojoj su spremljene visine od n učenika u razredu (ali su spremljene od najmanje prema najvećoj), a zatim unosi visinu x i provjerava nalazi li se visina x među unesenim visinama. Program će vratiti `True` ako se x nalazi u listi i `False` ako se ne nalazi zadana visina u listi.

Binarno pretraživanje

- radi samo na sortiranim listama
- puno brže od linearnog
- NE prolazi kroz sve elemente liste već provjerava je li zadovoljen neki uvjet

```
L = []
n = int(input())
for i in range(n): #kreiranje liste
    v = int(input())
    L.append(v)
print(L)
x = int(input()) #tražena visina
d = 0           #donja vrijednost indeksa
g = n-1        #gornja vrijednost indeksa
element = False #provjerava je li element u listi
while d<=g:
    s = (d+g)//2
    if x == L[s]:
        element = True
        break
    elif x < L[s]:
        g = s - 1
    else:
        d = s + 1
print(element)
```