LISTE U PYTHON-U

LISTE

- najčešće korišteni složeni tip podataka
- u listu smještamo međusobno povezane podatke
- elementi se nalaze u uglatim zagradama
- svakom elementu u listi možemo pristupiti navodeći njegov indeks na isti način kao i kod stringova

PRIMJERI LISTE I INDEKSIRANJA

```
L = [2, 5, 4, 7, 2]
L[0]
2
L[-2]
7
L[3:4]
[7]
L[2:]
[4, 7, 2]
L[10]#greška jer nema tog elementa
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    L[10]#greška jer nema tog elementa
IndexError: list index out of range
```

KREIRANJE LISTE

- nabrajanjem elemenata unutar uglatih zagrada
- primjenom naredbe for i funkcije range()
- iz neke postojeće zbirke, npr. iz stringa
- dodavanjem jednog po jednog elementa pomoću operatora + ili metode .append()
- prazna lista – prazne uglate zagrade ili funkcija list()

PRIMJERI KREIRANJA LISTE

```
L1 = [-4, 7, 3, 3.45, 0]
L1
[-4, 7, 3, 3.45, 0]
L2 = [i + 1 for i in range(5)]
L2
[1, 2, 3, 4, 5]
s = "auto"
L3 = list(s)
L3
['a', 'u', 't', 'o']
L4 = list()
L4
[]
```

```
L4 = L4 + [5] + [2]
L4
[5, 2]
L4.append(-1)
L4
[5, 2, -1]
L5 = []
L5
[]
L6 = 4*[0]
L6
[0, 0, 0, 0]
```

OPERATORI I FUNKCIJE NA LISTAMA

- možemo se koristiti osnovnim operatorima +, *, in te relacijskim operatorima
- određeni element liste možemo obrisati koristeći del
- možemo koristiti osnovne funkcije na listama – len, min, max

PRIMJERI UPOTREBE

```
L = [4, 7, 3, 5]
L + [1, 3]
[4, 7, 3, 5, 1, 3]
2*L
[4, 7, 3, 5, 4, 7, 3, 5]
del L[2]
L
[4, 7, 5]
len(L)
3
min(L)
4
max(L)
7
```

METODE ZA RAD S LISTAMA

metoda	opis djelovanja
`a.append(x)`	dodaje *x* na kraj liste *a*
`a.insert(i, x)`	dodaje element *x* prije *i*-tog elementa liste *a*
`a.index(x)`	vraća poziciju prvog pojavljivanja elementa *x* u listi *a*
`a.remove(x)`	izbacuje iz liste *a* element s vrijednosti *x*; ako takvih elemenata ima više, izbacit će onaj s najmanjim indeksom
`a.pop([i])`	vraća i izbacuje element liste koji se nalazi na poziciji *i*; ako *i* nije zadan, vraća i izbacuje zadnji element liste
`a.reverse()`	okreće listu tako da element koji je bio zadnji postaje prvi, predzadnji element postaje drugi itd.

PRIMJER UPOTREBE

```
L = [4, 5, 7, 9]
L.append(2)
L
[4, 5, 7, 9, 2]
L.index(7)
2
L.remove(9)
L
[4, 5, 7, 2]
L.pop(1)
5
L
[4, 7, 2]
L.reverse()
L
[2, 7, 4]
sorted(L)
[2, 4, 7]
L
[2, 7, 4]
L.sort()
L
[2, 4, 7]
```

→funkcija sorted kreira novu sortiranu listu dok je počenta ista, a metoda .sort() početnu listu sortira

METODE IZ MODULA RANDOM

funkcija	opis djelovanja
`randint(m, n)`	vraća nasumičan prirodni broj između *m* i *n*, uključujući i *m* i *n*
`randrange(m, n)`	vraća nasumičan prirodni broj između *m* i *n* – 1
`random()`	vraća nasumičan decimalni broj iz intervala [0, 1>
`choice(a)`	vraća jedan nasumični element liste *a*
`sample(a, n)`	vraća listu koja će biti načinjena od *n* nasumično odabranih elemenata liste *a*; *n* mora biti manji od duljine liste *a*; u dobivenoj listi svaki element liste *a* pojavit će se najviše jednom
`shuffle(a)`	nasumično razmješta elemente liste *a*; ova funkcija ništa ne vraća već će elementi liste *a* biti nasumično ispremještani

PRIMJER 1.

Napišite program koji će kreirati listu od n prirodnih brojeva te izračunati njihov zbroj i prosjek.

Ulaz:

L = [1, 6, 18, 25, 10]

Izlaz:

60

RJEŠENJE 1. PRIMJERA – PRVI NAČIN

```
n = int(input())
L = []
for i in range(n):
    x = int(input())
    L.append(x) #možemo koristiti i L = L + [x]
print(sum(L))
```

RJEŠENJE 1. PRIMJERA – DRUGI NAČIN

```
n = int(input())
L = [int(input()) for i in range(n)]
print(sum(L))
```

PRIMJER 2.

Napišite program koji će kreirati listu od 5 slučajnih brojeva između 10 i 20 te izračunati njihov prosjek.

Ulaz:

L = [11, 15, 17, 10, 13]

Izlaz:

13.2

RJEŠENJE 2. PRIMJERA

```
from random import*

L = [randint(10, 20) for i in range(5)]
print(sum(L)/5)
```

PRIMJER 3.

Napišite program koji će kreirati lisu od n natjecatelja u trčanju u koju spremamo njihovo vrijeme trčanja te odredite koji je po redu bio najbrži trkač.

Ulaz:

n = 5

L = [20.7, 21, 20.3, 19.7, 23]

Izlaz:

4.

RJEŠENJE 3. PRIMJERA

```
n = int(input())
L = [float(input()) for i in range(n)]
print(L.index(min(L))+1)
```

PRIMJER 4.

Napišimo program koji će unositi prirodan broj n – broj učenika u nekom razredu, a potom visine svih učenika u centimetrima. Program treba ispisati koliko je učenika iznadprosječno visoko.

Ulaz:

n = 5

L = [155, 167, 188, 173, 168]

Izlaz:

2

RJEŠENJE 4. PRIMJERA

```
n = int(input())
L = [int(input()) for i in range(n)]
prosjek = sum(L)/n
br = 0
for i in range(n):
    if L[i]>prosjek:
        br += 1
print(br)
```

PRIMJER 5.

Napiši program koji kreira listu od n elemenata i briše sve duplikate iz te liste te ispisuje novu listu bez duplikata.

Ulaz:

L = [11, 15, 11, 10, 13, 15]

Izlaz:

L = [15, 11, 10, 13]

RJEŠENJE 5. PRIMJERA

```
n = int(input())
L = [int(input()) for i in range(n)]
L2 = []
for i in range(n):
    if L[i] not in L2:
        L2.append(L[i])
print(L2)
```

PRIMJER 6.

Napiši program koji unosi bodove pet najboljih natjecatelja u skoku u dalj te ispisuje koliko je bodova imao drugi najbolji natjecatelj.

Ulaz:

L = [44, 78, 65, 71, 54]

Izlaz:

71

RJEŠENJE 6. PRIMJERA

```
L = [int(input()) for i in range(5)]
L.remove(max(L))
print(max(L))
```

ZADACI ZA VJEŽBU

1. Lista a zadana je naredbom: a = [2, 4, 6, 8, 9, 11]. Što će ispisati sljedeće naredbe?

a) print(a[3])

b) print(a[4:])

2. Napiši elemente liste a ako je lista zadana naredbom:

a) a = [i for i in range(5)]

b) a = [i + 2 for i in range(7)]

c) a = [i for i in range(10, 20, 3)]

d) a = [chr(i + 65) for i in range(8)]

e) a = [i ** 2 for i in range(2, 20, 4)]

3. Napiši naredbu kojom ćeš generirati listu:

a) svih neparnih brojeva do 20,

b) svih malih slova engleske abecede,

c) prvih deset parnih brojeva,

d) kubova prvih osam višekratnika broja 3.

4. Zadana je lista a = [3, 19, 4, 7, 8]. Ispiši sve elemente liste a nakon sljedećih naredbi (naredbe se izvršavaju po redu):

a) a.append(11)

b) a.pop()

c) a.remove(4)

d) a.insert(3, 6)

e) a.sort()

f) a.reverse()

5. Što će ispisati sljedeći niz naredbi?

```
s = 'krokodil'
l = s.split('o')
print(l)
```

6. Napiši program koji će unositi prirodan broj n, a zatim listu od n elemenata. Program treba ispisati elemente te liste zdesna u lijevo (posljednji uneseni element ispisuje se prvi).

Napomena: U ovom se zadatku svaki element liste unosi u svoj red, a u primjeru su elementi napisani u istom redu samo zbog estetskih razloga.

Ulaz:

n = 5

7 9 8 3 4

Izlaz:

4

3

8

9

7

7. Napišite program koji će u listu spremiti n prirodnih brojeva i kreirati novu listu koja sadrži samo parne brojeve.

Ulaz:

n = 5

L = [7, 8, 2, 4, 3]

Izlaz:

L2 = [8, 2, 4]

8. Napišite program koji će u listu spremiti n prirodnih brojeva i te odrediti zbroj samo onih u listi koji imaju zadnju znamenku 3.

Ulaz:

n = 5

L = [12, 24, 33, 2, 3]

Izlaz:

Zbroj = 36

9. Napišite program koji će u listu spremiti n cijelih brojeva te sve negativne brojeve pretvoriti u suprotne pozitivne.

Primjer:

Ulaz:

n = 5

L = [-1, 4, 3, -5, 7]

Izlaz:

L = [1, 4, 3, 5, 7]

10. Napiši program koji će unositi prirodan broj n, a zatim n prirodnih brojeva većih od 1. Program treba ispisati one od unesenih brojeva koji su prosti.

Ulaz:

5

7 9 8 3 4

Izlaz:

7

3

11. Napiši program koji će unositi prirodan broj n, a zatim dvije liste od po n elemenata. Obje liste trebaju biti sortirane uzlazno. Program treba generirati novu listu koja će se dobiti spajanjem unesenih lista i bit će isto tako sortirana uzlazno. (zadatak riješite pomoću sort i bez sort)

Ulaz:

5

3 5 11 19 25

6 10 11 20 26

Izlaz:

3 5 6 10 11 11 19 20 25 26

12. Dvije liste jednakih duljina možemo skalarno množiti tako da zbrajamo umnoške elemenata s istim indeksima. Ako su zadane liste a = [a1, a2, ..., an] i b = [b1, b2,..., bn], tada je skalarni produkt ovih dviju lista dan sa

$a \cdot b = a1 \cdot b1 + a2 \cdot b2 + \ldots + an \cdot bn.$

Napiši program koji će unositi prirodan broj n, a zatim dvije liste od po n elemenata i ispisivati njihov skalarni umnožak.

Ulaz:

N = 5

6 6 5 7 2

8 3 6 3 2

Izlaz:

121

13. Antonio u spremniku svog automobila ima goriva za n kilometara. Njegov je posao da obilazi gradove i dostavlja pakete turističkim uredima. Kako je plaćen po obiđenom gradu, htio bi s gorivom s kojim raspolaže obići što više gradova. Svi se gradovi nalaze duž ravne ceste koja počinje u nultom kilometru, a može biti dugačka najviše 1000 kilometara. Antonio vožnju započinje u gradu koji je t kilometara udaljen od početka ceste. On točno zna na kojem se kilometru od početka ceste nalazi koji grad. Antonio na samom početku odabire smjer u kojem će krenuti dijeliti pakete. Pomogni Antoniju i napiši program koji će odrediti u kojem smjeru treba krenuti kako bi obišao što više gradova, odnosno koliko najviše gradova može obići s količinom goriva s kojom raspolaže.

Napiši program koji će unositi:

prirodni broj n – broj gradova duž ceste

udaljenosti svakog pojedinog grada od početka ceste

udaljenost mjesta iz kojeg Antonio kreće od početka ceste

najveći broj kilometara koji Antonio može prijeći s gorivom koje ima u spremniku.

Program treba ispisati najveći mogući broj gradova koje Antonio može obići.

Ulaz:

6

1 3 8 9 18 26

9

7

Izlaz:

3