

LISTE – zadaci (sortiranje, parsiranje, stack, queue, lista listi)

Happy Hour



Three programmers walk into a bar... Cool Python Codes

Metode za rad sa listama

`lista.append (objekt)` -dodaje dati objekt na kraj liste

`lista.count (objekt)` -vraća broj pojavljivanja zadanoga objekta u listi

`lista.index (objekt)` -vraća prvi indeks na kojemu se pojavljuje zadani objekt u listi

`lista.insert (i, objekt)` -dodaje objekt u listu na zadanu poziciju i

`lista.pop (i)`-vraća objekt na i-toj poziciji u listi ili objekt na poslednjoj poziciji ako indeks nije naveden i briše taj objekt iz liste

`lista.remove (objekt)` -briše prvi objekt iz liste koji je jednak zadanomu objektu

Zadatak 01:

Napisati program koji će učitati prirodan broj $N < 100$, a potom i listu celih brojeva sa N elemenata. U listi brojeva odredi najveći broj strogo manji od maksimalne vrednosti i ispisati taj broj. Ako nema takve vrednosti ispiši -1.

Rešenje:

```
#print('Unesite broj N ')
N=int(input())
L=[]
#print('Unesite ',N,' članova liste: ')
for i in range(N):
    x=int(input())
    L=L+[x]
M = max(L)
while L.count(M) > 0:
    L.remove(M)

if len(L) != 0:
    print(max(L))
else:
    print(-1)
```

Zadatak 02:

https://petlja.org/biblioteka/r/problemi/Zbirka/sortiranje_brojeva

Napiši program koji uređuje (sortira-algoritmom Tim sort) niz brojeva neopadajuće (svaki naredni mora da bude veći ili jednak od prethodnog).

Rešenje:

```
n = int(input())
brojevi = [int(input()) for i in range(n)]
brojevi = sorted(brojevi)
for i in brojevi:
    print(i)
```

Zadatak 03:

Sortirati niz iz prethodnog zadatka u opadajućem poretku

Rešenje

```
n = int(input())
brojevi = [int(input()) for i in range(n)]
brojevi = sorted(brojevi, reverse=True)
for i in brojevi:
    print(i)
```

Zadatak 04:

https://petlja.org/biblioteka/r/problemi/Zbirka/vrednost_izraza

Napiši program koji izračunava vrednost jednostavnog aritmetičkog izraza (npr. $123*456$). Sa standardnog ulaza učitava se jedna linija teksta koja sadrži jednostavni aritmetički izraz dobijen primenom operacija sabiranja, oduzimanja ili množenja na dva prirodna broja. Učitani tekst ne sadrži razmake. Napomena: vežbe radi, učitati ceo string koji sadrži izraz, a zatim ga raščlaniti na delove.

Rešenje :

Ako string `s` sadrži učitani izraz, potrebno je raščlaniti ga na dva operanda i operator koji se nalazi između njih.

Prvi zadatak je naći poziciju operatora. Jedan način da se to uradi je da se upotrebi biblioteka funkcija. U jeziku Python to može biti metoda `index` (kao u resenju 2).

Naravno, umesto bibliotekskih funkcija pretragu je moguće implementirati i ručno (korišćenjem algoritma linearne pretrage, poput, resenja 1), mada je rešenje uz upotrebu bibliotekskih funkcija svakako bolje.

Nakon pronalaženja pozicije `p` na kojoj se nalazi operator, potrebno je izdvojiti deo stringa ispred operatora (on sadrži prvi operand) i deo stringa iza operatora.

To je opet moguće učiniti bibliotekskim funkcijama.

Mi smo ipak u oba resenja metod za izdvajanje podstringa smo implementirali ručno. Potrebno je željeni raspon karaktera kopirati iz jednog stringa u drugi.

Napokon dolazimo do centralnog pitanja u zadatku a to je kako od niza karaktera (tj. stringa) koji sadrži isključivo cifre doći do vrednosti broja. Opet nam na raspolaganju stoje biblioteka funkcije tj. transformacija `int` metodom.

Ručna implementacija transformacije stringa u broj zasnovana je na Hornerovoj šemi (I to nismo radili ni u resenju 1, niti u resenju 2).

RESENJE 1:

```
def pozicijaOperatora(s):  
    for i in range(len(s)):  
        if s[i] in {'+', '-', '*'}:  
            return i  
    return -1  
  
s = input()  
poz = pozicijaOperatora(s)  
operand1 = s[0:poz]  
oper = s[poz]  
operand2 = s[poz+1:]  
a = int(operand1)  
b = int(operand2)  
if oper == '+':  
    rezultat = a + b  
elif oper == '-':  
    rezultat = a - b  
elif oper == '*':  
    rezultat = a * b  
print(rezultat)
```

RESENJE 2:

```
s = input()  
  
pozicijaOperatora = min(s.index(c) for c in "+-*" if c in s)  
operand1 = s[0:pozicijaOperatora]  
oper = s[pozicijaOperatora]  
operand2 = s[pozicijaOperatora+1:]  
a = int(operand1)  
b = int(operand2)  
if oper == '+':  
    rezultat = a + b  
elif oper == '-':  
    rezultat = a - b  
elif oper == '*':  
    rezultat = a * b  
print(rezultat)
```

Stek (Stack) je lista koja omogućuje dodavanje i izbacivanje elemenata poštujući LIFO pravilo, tj. Last-in, First-out.

```
stek = ['a', 'b', 'c', 'd']
```

```
stek.append('e')
```

```
stek.append('f')
```

```
print(stek)
```

```
print('Vrh steka je ', stek.pop())
```

```
print('Novi stek je ', stek)
```

Red (Queue) je lista koja omogućuje dodavanje i izbacivanje elemenata pri čemu poštuje FIFO pravilo, tj. First-in, First-out.

```
red = ['a', 'b', 'c', 'd']
```

```
red.append('e')
```

```
red.append('f')
```

```
print(red)
```

```
print(red.pop(0)) # skidanje slova 'a' iz FIFO reda
```

```
print(red) # stampa ['b', 'c', 'd', 'e', 'f']
```

Lista listi

```
L = [ [1, 2, 3], [4, 5, 6] ]
```

```
print(L[1][0]) # stampa 4
```

```
for i in L:
```

```
    for j in i:
```

```
        print(j, end = ' ') #stampa 2D listu kao 1D listu, row major order
```

```
print()
```

Učitavanje n celih brojeva u jednom redu u listu i štampanje sortirane liste u jednom redu

```
# učitavanje broja n i n celih brojeva u jednom redu
```

```
n = int(input())
```

```
brojevi = list(map(int, input().split()))
```

```
#sortiranje n brojeva
```

```
brojevi.sort(reverse=True)
```

```
#stampanje brojeva u jednom redu
```

```
for i in brojevi:
```

```
    print(i, end=" ")
```