PONAVLJAMO (ob koncu semestra)

1. Sestavi funkcijo `vsebovani(s, t)`, ki ugotovi, če so vsi znaki niza `s` vsebovani v nizu `t`. Klic `vsebovani('osa', 'postaja')` torej vrne True, `vsebovani('osa', 'proza')` pa `False`. **Neobvezni dodatek v izziv (loti se ga, ko rešiš vse ostalo)**: kaj, če zahtevamo, da so vsebovani v istem relativnem vrstnem redu. Torej `vseb_relativno('osa', 'postaja')` je True, `vseb_relativno('osa', 'opastja')`je False, `vseb_relativno('oso', 'opastja')` je False.

2. Sestavite funkcijo `karo(n)`, ki izriše karo iz znakov '*'.

```
>>> karo(4)
   *
  ***
 *****
*******
 *****
  ***
   *
```

```
>>> karo(3)
  *
 ***
*****
 ***
  *
```

3. Sestavi program, ki prebere nenegativno število `n` in izpiše n – 1 vrstic oblike `'i --- n-i'`, kjer je `i` zaporedna številka vrstice. Če vnesemo število 4, je ustrezen izpis

```
1---3
2---2
3---1
```

4. Napiši program, v katerega bo uporabnik vpisoval števila. Program bo sproti izpisoval produkt do takrat vnesenih števil. Ko bo produkt presegel število 100, se bo program ustavil in izpisal `'To je preveč!'`. Če uporabnik vpiše število 0, naj program izpiše 0 in konča. Primera:

```
Vnesi število: 5
5
Vnesi število: 3
15
Vnesi število: 7
To je preveč!
```

```
Vnesi število: 5
5
Vnesi število: 3
15
Vnesi število: 0
0
```

5. Napiši funkcijo `zdruzi(tabela tabel)`, ki sprejme tabelo tabel (npr. `[[1,2,3,4],[8,4],[3],[4,5,3,2],[3,4,8],[6,4]]`) in vrne tabelo z elementi tistih podtabel, ki vsebujejo vsaj tri elemente. Za zgornjo tabelo naj funkcija vrne tabelo `[1, 2, 3, 4, 4, 5, 3, 2, 3, 4, 8]`.

6. Lojze se je odločil, da bo s svojimi starimi zapiski iz študentskih let poskusil kaj zaslužiti. Našel je nekaj zainteresiranih kupcev. Vsak je na list zapisal svoje ime in koliko je pripravljen za zapiske odšteti, Lojze pa je nato zapiske prodal tistemu, ki je ponudil največ. Napiši program, ki prebere imena in ponudbe kupcev in vrne ime tistega, ki je največ ponudil. Če je več takih, ki so ponudili največ, naj vrne imena vseh. Program naj preneha z branjem podatkov, kadar za ime vnesemo niz `'Konec'`.

Če zaporedoma vnašamo podatke:

```
Jože
50
Franci
100
Janez
70
Tone
100
Konec
```

naj program izpiše:

```
Največ je ponudil (so ponudili):  Franci, Tone
```

7. Permutacijo podamo s slovarjem, kjer se vsako naravno število od 1 do n pojavi enkrat kot ključ in enkrat kot vrednost nekega ključa, npr. p = {1:4, 2:2, 3:6, 4:7, 5:8, 6:5, 7:1, 8:3}. Sestavi funkcijo, ki poišče in vrne cikel, ki se začne pri danem ključu (cikel podamo s seznamom, kjer po vrsti naštejemo, kako elementi prehajajo eden v drugega). Cikel, ki se v permutaciji p začne pri ključu 3, je [3, 6, 5, 8], cikel, ki se začne pri ključu 2, pa [2].

8. Napišite funkcijo `zadnjiLihi(s)`, ki kot argument prejme seznam števil `s`. Njihov vrstni red naj spremeni tako, da bodo v njem najprej soda, nato liha števila – vsaka zase v enakem vrstnem redu, kot so bila prej. Funkcija ne sme vračati ničesar. Če napišete funkcijo, ki vrne nov seznam, dobite največ 5 točk.

Primer ("polne rešitve")

```
>>> s = [5, 8, 4, 17, 13, 10, 9]
>>> zadnjiLihi(s)
>>> s
>>> [8, 4, 10, 5, 17, 13, 9]
>>> # 5, 17, 13 in 9 so v enakem vrstnem redu kot prej; soda tudi
```