



Osnove 6

Python urejevalnik se nahaja na strani:

<https://www.w3resource.com/python-exercises/python-basic-exercises.php#EDITOR>

Naloge Kattis: <https://github.com/minidomo/Kattis>

1. **Štiri naravna števila** Komisija tekmovanja v računalništvu je dobila a lahkih nalog, b srednje lahkih in c težjih nalog. Rada bi sestavila nabor n nalog, ki bi vseboval vsaj eno lahko eno srednje lahko in eno težjo nalogo.

Napiši program, ki bo iz standardnega vhoda prebral 4 naravna števila in na standardni izhod izpisal **da**, če je pogoj izpolnjen in **ne**, če pogoj ni izpolnjen.

2. **Ostanki pri deljenju** Dani sta dve naravni števili a in b , $\text{mod}(a, b)$ je funkcija, ki dvema naravnima številoma a in b določi kolik je ostanek pri deljenju števila a s številom b .

Napiši program, ki iz standardnega vhoda prebere 10 naravnih števil manjših od 100 na standardni izhod izpiše koliko smo dobili različnih ostankov pri deljenju teh števil z 42.

3. Pot

Vsak kolesar je dobil na lističih opis poti. Na Vsakem lističu je opisana ena od etap na poti. Na prvem mestu je zapisan kraj začetka etape in kraj cilja etape. Vsak kolesar je dobil lističe urejene v pravilnem zaporedju, kot so si etape sledile. Cilj ene etape na enem lističu je start naslednje etape na naslednjem lističu.

Pohodnik Niko Raztresen je po nesreči pomešal lističe, in sedaj ne ve, kje začeti pot. Prosi vas, da mu lističe spet uredite po vrsti.

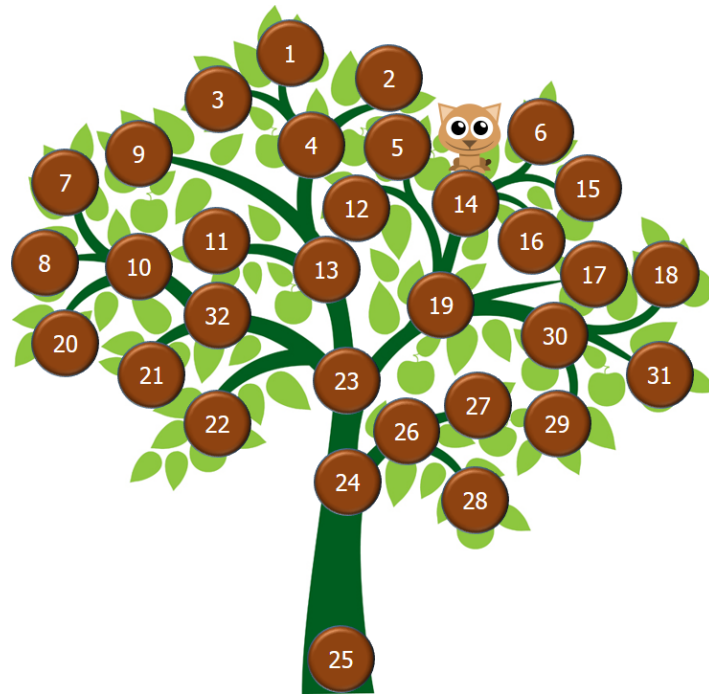
Vhod: Iz datoteke preberemo vrstice na katerih je zapisan začetek in cilj posamezne etape.

Izhod: Na standardni izhod izpiši vrstice v pravilnem vrstnem redu.

Primer:

vhod:	izhod:
Vrhnika Logatec	Ljubljana Vrhnika
Štanjel Hruševica	Vrhnika Logatec
Ljubljana Vrhnika	Logatec Podkraj
Podkraj Vrhpolje	Podkraj Vrhpolje
Logatec Podkraj	Vrhpolje Štaniel
Hruševica Kosovelje	Štanjel Hruševica
Vrhpolje Štaniel	Hruševica Kosovelje

4. **Muca na drevesu** Muca se je zataknila na drevesu. Spravite jo iz drevesa tako, da ji poveste zaporedje števil v razvejiščih po katerih se mora spustiti na tla.



Vhod: Na vhodu preberemo opis drevesa iz datoteke.

V prvi vrstici je podano razvejišče, kjer se nahaja muca. V drugi vrstici je številka korena in številka prvega razvejišča na deblu. V ostalih vrsticah je na prvem mestu zapisano razvejišče, ki mu sledijo številke razvejišč na vejah, ki izraščajo iz njega.

Izhod: Na standardni izhod zapišemo zaporedje razvejišč, ki bodo muco pripeljali na tla.

Primer na sliki:

Vhod:

```
14
25 24
4 3 1 2
13 9 4 11
10 20 8 7
32 10 21
23 13 19 32 22
19 12 5 14 17 30
14 6 15 16
30 18 31 29
24 23 26
26 27 28
```

Izhod:

```
14 19 23 24 25
```

5. **Igra s faktorji** Igrata dva igralca.

a) Prvi izbere naravno število označimo ga z n .

b) Drugi pa poskuša poiskati k naravnih števil m_1, \dots, m_k , tako da je:

$$(m_1 + 1)(m_2 + 1) \cdots (m_k + 1) = n$$

Drugi igralec dobi toliko točk, kolikor je faktorjev. V našem primeru je to k .

Napišite program, ki bo igral v vlogi drugega igralca.

Vhod: Iz standardnega vhoda preberete naravno število $10^3 \leq n \leq 10^6$.

Izhod: Na standardni izhod izpišete število k .

Primeri:

65536 -> 16

127381 -> 3

6. **Dva stolpa** Imamo dve skupini s tremi kvadri z enakimi osnovnimi ploskvami in različnih višin. Za vsako skupino vemo, kolika je višina stolpa, ko postavimo kvadre enega na drugega.

Po nesreči so se kvadri pomešali. Vaša naloga, da kvadre spet razdelite v dve skupini tako, da bo višina stolpov v skupinah taka, kot je bila na začetku.

Predpostavili bomo, da so višine različne in da je rešitev enolična.

Vhod: Na vhodu preberemo iz standardnega vhoda osem vrednosti. Prvih šest predstavljajo višine pomešanih kvadrov, vrstni red je poljuben sedma in osma vrednost predstavlja predpisano višino stolpov v vsaki od skupin.

Izhod: Na standardni izhod izpišite šest vrednosti prve tri za prvo skupino in druge tri za drugo skupino.

Primeri:

12 8 2 4 10 3 25 14 ---> 12 10 3 8 4 2

12 17 36 37 51 63 92 124 ---> 63 17 12 51 37 36

Rešitve 6

Štiri naravna števila

```
def stiri(a, b, c, n):  
    return (a >= 1) and (b >= 1) and (c >= 1) and ((a + b + c) >= n) and n => 3
```

Pot

```
#!/usr/bin/env python3  
  
seznam = '''Vrhnika Logatec  
Štanjel Hrušica  
Ljubljana Vrhnika  
Podkraj Vrhopolje  
Logatec Podkraj  
Hrušica Kosovelje  
Vrhopolje Štanjel'''  
  
def pot(seznam):  
    etape = dict([x.split() for x in seznam.split("\n")])  
    kljuci = set(etape.keys())  
    vredno = set(etape.values())  
    start = (kljuci - vredno).pop()  
    tekoci = start  
    res = []  
    while etape.get(tekoci):  
        naslednji = etape.get(tekoci)  
        res.append((tekoci, naslednji))  
        tekoci = naslednji  
    return res  
  
if __name__ == '__main__':  
    seznam = pot(seznam)  
    for x in seznam:  
        print(x[0] + ' ' + x[1])
```

Muca

```
#!/usr/bin/env python3

drevo = '''14
25 24
4 3 1 2
13 9 4 11
10 20 8 7
32 10 21
23 13 19 32 22
19 12 5 14 17 30
14 6 15 16
30 18 31 29
24 23 26
26 27 28'''

if __name__ == '__main__':
    #with open(file_name, 'r') as file:
    #poti = file.read().split('\n')

    poti = drevo.split('\n')
    poti = [x.split() for x in poti]
    poti = {x[0]:x[1:] for x in poti}
    ip = iter(poti)
    dol = []
    rv = next(ip)
    dol.append(rv)
    while True:
        for r in poti:
            test = False
            if rv in poti[r]:
                rv = r
                dol.append(rv)
                test = True
                break
        if not test:
            break

    print(dol)

if __name__ == '__main__':
    #print(parse_drevo('muca.txt'))
```

Igra s faktorji

```
#!/usr/bin/env python3

def faktorizacija(n):
    fact = []
    i = 2
    while (i <= n):
        if n % i == 0:
            fact.append(i)
            n //= i
        else:
            i += 1
    return fact

if __name__ == '__main__':
    n = int(input('n -> '))
    print(len(faktorizacija(n)))
```

Dva stolpa

```
#!/usr/bin/env python3

def stolpa(x, h):
    for i in range(len(x)):
        a = x[i]
        del x[i]
        for j in range(len(x)):
            b = x[j]
            del x[j]
            c = h[0] - (a + b)
            if c in x:
                x.remove(c)
                return [a, b, c], x
        x.insert(j, b)
    x.insert(i, a)

if __name__ == '__main__':
    l = [int(x) for x in input(' --> ').split()]
    n = len(l)
    h = l[-2:]
    x = l[:-2]
    print(stolpa(x, h))
```