



Osnove 3

Python urejevalnik se nahaja na strani:

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

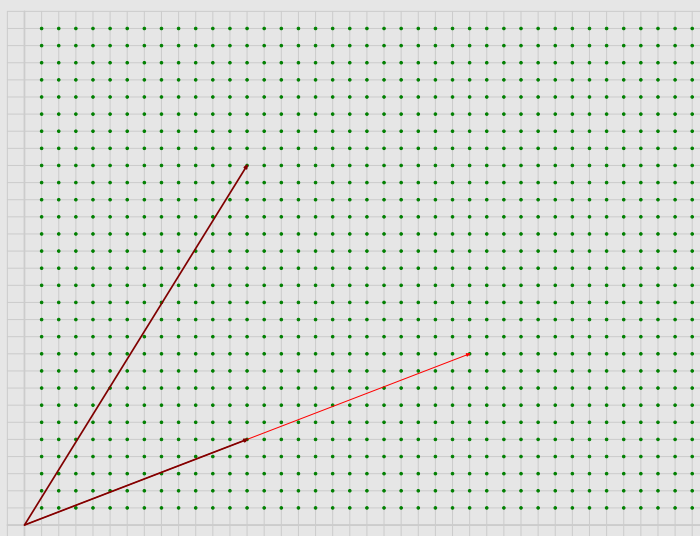
1. **Oklepaji** Napišite program, ki bi nam pomagal ugotoviti napako pri grupiranju v aritmetičnem izrazu, kjer se uporabljajo tri vrste oklepajev. Okrogli $()$, oglati $[]$ in zaviti $\{\}$. Ostale sintaktične napake pri pisanju aritmetičnega izraza nas ne zanimajo. Prioritete oklepajev ne upoštevamo. To pomeni, da zaporedje uporabljenih oklepajev ni pomembno. Primeri:

$\{2+4*[1-3]+(2-4)*\{2-3*[\{8+3\}*(3+5)]\}\}$ --> Pravilno

$\{2+4*[1-3]+(2-4)*2-3*[\{8+3\}*(3+5)]\}$ --> Napačno

2. **Nasad** Nasad jablan je v obliki pravokotnika in v njem so jablane zasajene v enakomernih presledkih. Štiri sosednje jablane tvorijo oglišča kvadrata s stranico $a = 1$. Jablane so zasajene v $M = 600$ stolpcih in $N = 700$ vrsticah. Izhodišče koordinatnega sistema in koordinatne osi postavimo tako, da je jablana v spodnjem levem oglišču postavljena v točki $T_0(1, 1)$, medtem ko je jablana v nasprotnem oglišču pravokotnika postavljena v točki $T_1(M, N)$. V točki $O(0, 0)$ se nahaja laser. Laser usmerimo naključno proti eni od jablan. Vzemimo jablano, postavljeno v točki $T(i, j)$, kjer sta i in j naravni števili $1 \leq i \leq M$ in $1 \leq j \leq N$. V nekaterih primerih žarek ne bo dosegel dane jablane, ker ga bo zasenčila jablana, ki je bližje laserju. V drugih primerih pa laserski žarek doseže zeleno jablano. Iz standardnega vhoda preberite koordinati jablane i in j . Na koncu izpišite koordinati (x, y) jablane, ki jo bo žarek zadel.

Slika



Primer

1. Vhod: 13 21 2. Vhod: 26 10
1. Izhod: 13 21 2. Izhod: 13 5

3. **Prostornina stolpa** Zgradimo stolp iz n kock. Stranica kocke na dnu je enaka n enot, kocka, ki ji sledi, ima stranico $n - 1$ enot in tako naprej vse do vrha, ko ima zadnja kocka stranico enako 1 enoto. Koliko je prostornina stolpa? Iz standardnega vhoda preberite naravno število n in na standardni izhod izpišite prostornino stolpa.

Primer

Vhod: 27 Izhod: 142884

4. **Bisekcija** Ciljate točko p , ki se nahaja nekje med 264 in 480 metri od topa. Funkcija `bum` simulira top. Parameter, ki ga sprejema, je količina smodnika. Vemo, da je količina `smodnik = 10` premajhna, krogla pristane v vsakem primeru pred ciljem, in da je količina `smodnik = 16` prevelika, krogla pristane za ciljem. Izbrati morate programsko količino smodnika, ki bo ravno pravšnja, da bo krogla zadela cilj. Pri vsakem zagonu programa se položaj točke p izbere naključno. Vzeli bomo, da je cilj zadet, če krogla pade na mesto, ki je oddaljeno za manj ko meter od cilja.

Izpišite vrstice s količinami smodnika pri posameznih poskusih. Prvi stolpec je položaj točke p , drugi stolpec so mesta, kjer je pristala krogla, in v zadnjem stolpcu so količine smodnika.

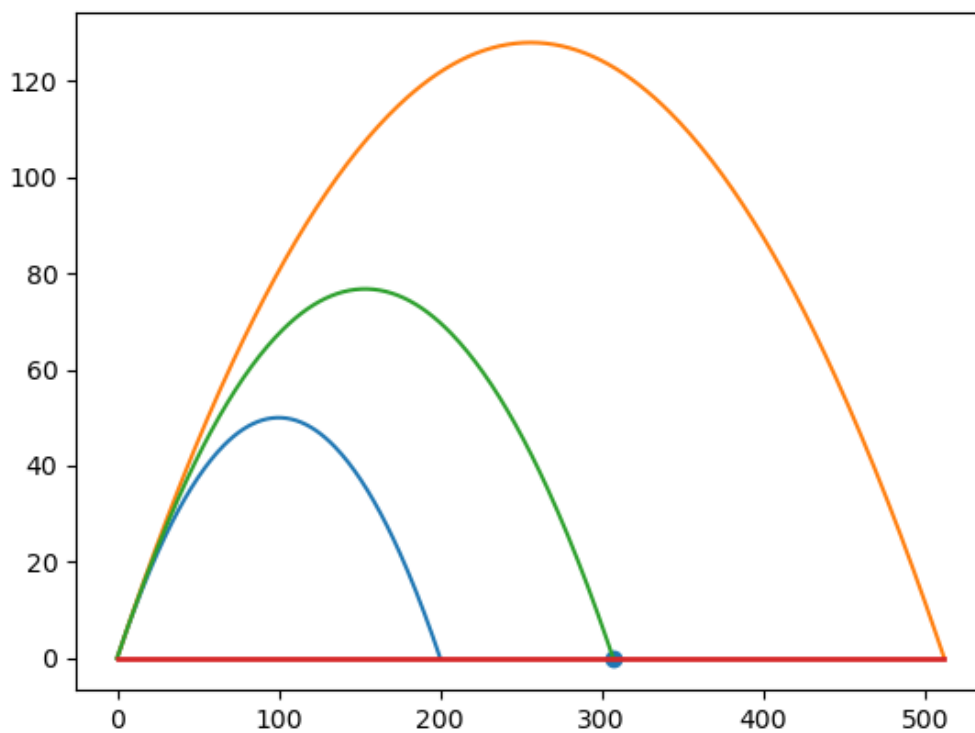
Procedura `bum`

```
import random
p = random.randint(264, 480)
def bum(smodnik):
    x, y, g = 0, 0, 1
    dt = 0.1
    vx = vy = smodnik
    while y >= 0:
        x += vx*dt
        vy -= g*dt/2
        y += vy*dt
        vy -= g*dt/2
    return x
```

Primer

Izhod:

```
307 338 13.00
307 264 11.50
307 300 12.25
307 319 12.62
307 309 12.44
307 304 12.34
307 307 12.39
```



Rešitve 3

Oklepaji

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

b = {"{":"}", "[":"]", "(":")"}
sklad = []

def test(line):
    for x in line:
        if x in b.keys():
            sklad.append(x)
        elif x in b.values():
            try:
                y = sklad.pop()
                if x != b[y]:
                    return False
            except:
                return False
    return True

if __name__ == '__main__':
    line = input("Izraz -> ").split()
    print(line)
    print(test(line))
```

Nasad

```
def gcd(m,n):
    if m<=0 or n<=0:
        return(0)
    if m < n:
        m, n = n, m
    while n != 0:
        m, n = n, m % n
    return m

if __name__ == '__main__':
    M=600
    N=700
    s=input('Vpiši cilj: m n --> ')
    c=[int(x) for x in s.split(' ')]
    g=gcd(*c)
    c=[x//g for x in c]
    if (0<c[0] and c[0]<=M) and (0<c[1] and c[1]<=N):
        print(c)
    else:
        print([0,0])
```

Prostornina stolpa

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

def vol(n):
    s = sum([x**3 for x in range(1,n+1)])
    return s

if __name__ == '__main__':
    n = int(input('Vpiši višino stolpa -> '))
    print(vol(n))
```

Bisekcija

```
#!/usr/bin/env python3
import random
import matplotlib.pyplot as plt

p = random.randint(264, 480)

def bum(smodnik):
    x, y, g, dt = 0, 0, 1, 0.1
    xx, yy = [x], [y]
    vx = vy = smodnik
    while y >= 0:
        x += vx*dt
        vy -= g*dt/2
        y += vy*dt
        vy -= g*dt/2
        xx.append(x)
        yy.append(y)
    return x, xx, yy

def bisekcija():
    a, b = 10, 16
    while True:
        m = (a + b)/2
        q, xx, yy = bum(m)
        print('%d %d %0.2f' %(p, q, m))
        if abs(q - p) < 1:
            return m, xx, yy
        if q < p: a = m
        elif q > p: b = m
        else:
            return m, xx, yy

if __name__ == '__main__':
    q1, x1, y1 = bum(10)
    q2, x2, y2 = bum(16)
    q, x, y = bisekcija()
    plt.plot(x1, y1)
    plt.plot(x2, y2)
    plt.plot(x, y)
    plt.plot([0,x2[-1]], [0,0], linewidth=2)
    plt.scatter([p], [0])
    plt.show()
```