

Uravnotežen trojiški sestav

Naloga

Številski sestavi določajo, kako se zapisujejo števila. Mi uporabljamo desetiški številski sestav. Osnova tega sestava je $b = 10$. Števk, znakov, s katerimi zapisujemo števila, je tudi 10. To so 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Njihove številске vrednosti so 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Velja

$$53107_{10} = 5 \cdot 10^5 + 3 \cdot 10^4 + 1 \cdot 10^3 + 0 \cdot 10^2 + 7 \cdot 10^0.$$

V mestnem zapisu je zelo pomembno, na katerem mestu stoji posamezna številka. Manjkajoče potence števila 10 v razvoju napovemo s številko 0.

Ogledali si bomo poseben številski sestav. V osnovi je to trojiški sestav. V običajnem trojiškem številskem sestavu potrebujemo tri številke 0, 1, 2, katerih številске vrednosti so enake 0, 1, 2. Zapis 210₃ v tem sestavu predstavlja število

$$210_3 = 2 \cdot 3^3 + 1 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0 = 2 \cdot 27 + 9 + 1 = 54 + 9 + 1 = 64_{10}.$$

Vendar pa je naš trojiški sestav drugačen. Koeficienti pred potencami števila 3 niso 0, 1 in 2, ampak -1, 0 in 1. Številke bomo označili z znaki -, 0, +, številске vrednosti pa so enake -1, 0, 1. Zapis +00-00 predstavlja število

$$+00-00 \rightarrow 1 \cdot 3^5 + 0 \cdot 3^4 + 0 \cdot 3^3 - 1 \cdot 3^2 + 0 \cdot 3^1 + 0 \cdot 3^0 = 234_{10}.$$

Naj omenimo, da je $3^0 = 1$. Če v zapisu zamenjamo med seboj znaka - in +, dobimo ustrezen zapis števila, pomnoženega z -1.

Vhod:

Iz standardnega vhoda preberete niz znakov +, 0 in -.

Izhod:

Na standardni izhod izpišete ustrezno število v desetiškem zapisu.

Primer:

```
vhod: +00-00   izhod 234
vhod: -00+00   izhod -234
```

Tarča

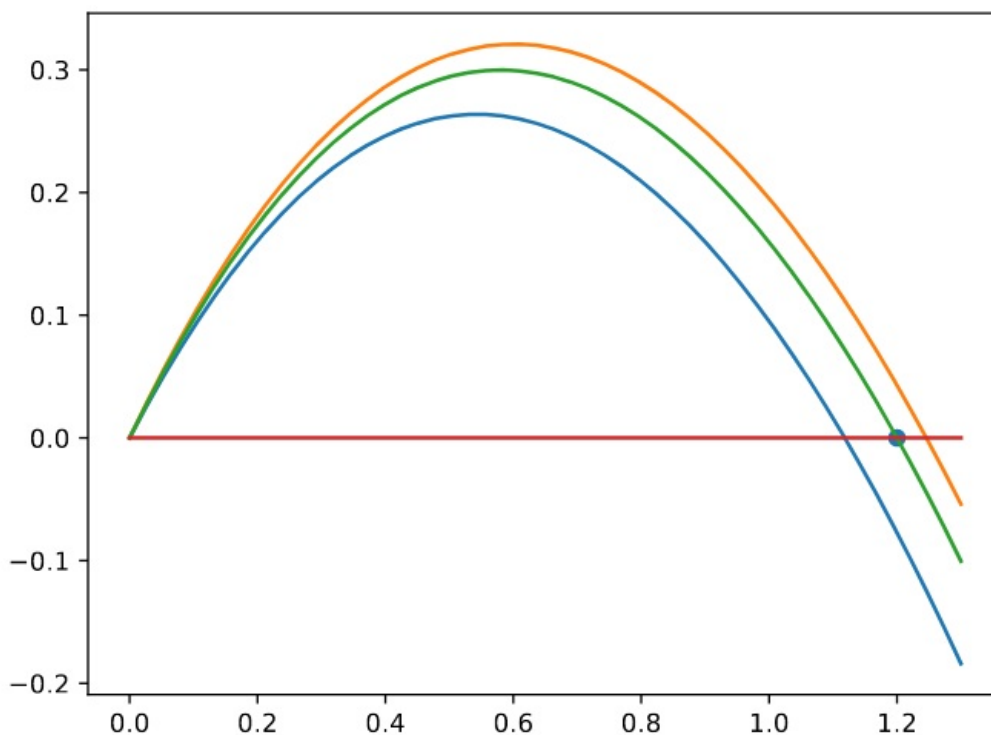
Naloga

Dana je funkcija $f(a, x)$, odvisna od parametra a . Graf funkcije predstavlja trajektorijo izstrelka. Top se nahaja v koordinatnem izhodišču. Tarča je točka $x_0 = 1.2$ na osi x . Izberi parameter a tako, da bo izstrelak zadel tarčo. Vemo, če je $a_1 = 1$, pade izstrelak pred tarčo, medtem ko za $a_2 = 1.1$ pade izstrelak za tarčo.

```
import numpy as np
def f(x, a):
    return a * x - np.exp(-x / 10) * x**2
```

Funkcijo definirate v programu. Sama definicija funkcije vas ne zanima. Zanimajo vas samo njene vrednosti. Glejte jo kot črno škatlico.

Nato izberite vrednost parametra $a = a_0$, da bo $f(x_0, a_0) = 0$. Uporabite metodo bisekcije. Za dodatek zadenite še tarčo $x = 0.95$. Meji x_1 in x_2 določite sami.



Vhod:

Vhodne podatke definirate v programu. Ti so funkcija $f(x, a)$, parametra $a_1 = 1$ (zadetek pred tarčo), $a_2 = 1.1$ (zadetek za tarčo) in koordinata tarče $x_0 = 1.2$.

Izhod:

Na standardni izhod izpišete vrednost parametra a_0 . Ta je pri gornjih podatkih enak 1.064

Faktorizacija

Naloga

Dano naravno število razčlenite na produkt praštevil. Praštevilo je naravno število, ki je deljivo le z 1 in samim seboj. Števila 1 ne prištevamo med praštevila. Tako razčlenitev števila na produkt praštevil postane enolična.

Vhod:

Iz standardnega vhoda preberete število.

Izhod:

Na standardni izhod izpišete praštevila, urejena v naraščajočem vrstnem redu, ločena s presledkom.

Primeri:

```
vhod: 144  izhod: 2 2 2 2 3 3  
vhod: 17  izhod: 17
```

Vsota elementov gnezdenega seznama

Gnezdeni seznam je seznam seznamov. Elementi seznama so poleg osnovnih elementov tudi sezname. Primer gnezdenega seznama celih števil:

```
s = [[1,4,2,[4,5,[3,1]],3,1,3],4,9,2,[8,9,3,[1,2]]]
```

Naloga

Poiščite vsoto števil v takšnem gnezdenem seznamu. Program mora znati izračunati vsoto za čisto splošen gnezdeni seznam.

Vhod

Iz standardnega vhoda preberete seznam.

Izhod

Na standardni izhod izpišete vsoto elementov seznama.

Primer

```
Vhod: [[1,4,2,[4,5,[3,1]],3,1,3],4,9,2,[8,9,3,[1,2]]]  
Izhod: 65
```

Pari

Naloga

Poiščite pare naravnih števil iz danega seznama, katerih vsota je dano število.

Vhod:

Iz standardnega vhoda preberite seznam števil, kjer so števila ločena s presledki. Temu sledi ločilo vejica in število, ki pomeni vsoto.

Izhod:

Na standardni izhod izpišite pare, katerih vsota je dano število.

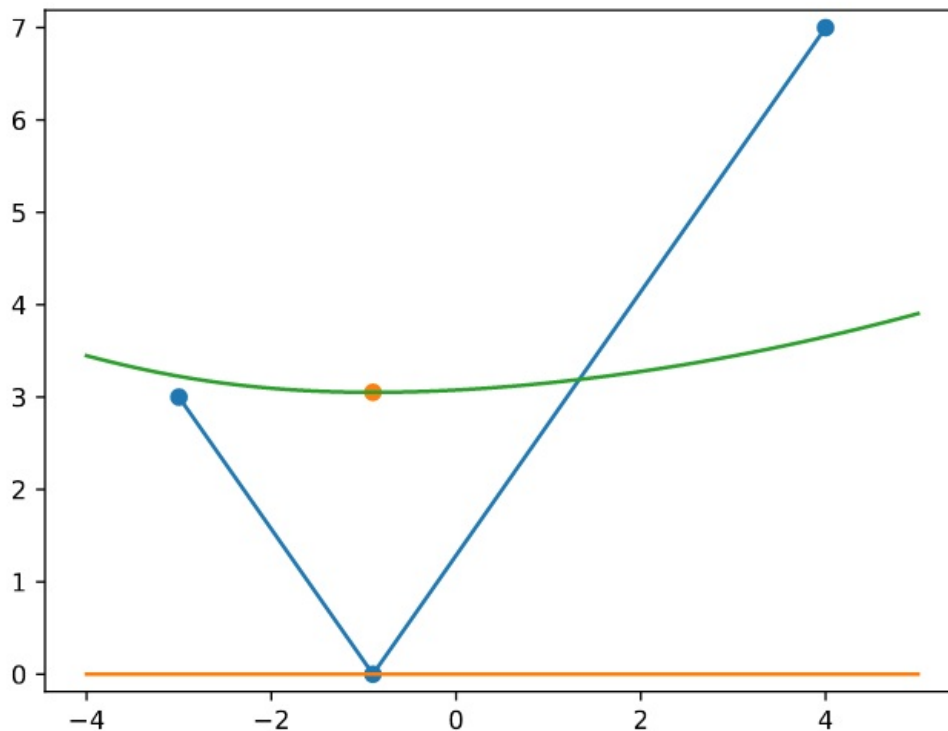
Primeri:

```
Vhod 1: 2 3 4 4 5 2 6 7 8 4, 8  
Izhod 1: [(2, 6), (3, 5), (4, 4)]
```

```
Vhod 2: 2 4 5 7 9, 11  
Izhod 2: [(2, 9), (4, 7)]
```

Pristanišče

Dve mesti, ki ležita nedaleč od obale, sta se odločili, da na obali postavita skupno pristanišče. Postaviti ga želita na tisto mesto na obali, da bo skupna dolžina poti od mest do pristanišča najkrajša.



Na karti je predstavljen načrt. Obala teče po osi x , na zgornji polovici je kopno, medtem ko je na spodnji polovici ravnine morje. Koordinate mest so $T_1(x_1, y_1)$ in $T_2(x_2, y_2)$. Koordinati x_1 in x_2 sta določeni v naprej $T_1 = (-3, y_1)$ in $T_2 = (4, y_2)$. Koordinati $y_1 > 0$ in $y_2 > 0$ preberemo iz standardnega vhoda. Točko, kjer se bo umestilo pristanišče $T_o = (x, 0)$, poiščemo s pomočjo trisekcije. Interval $[x_1, x_2]$ razdelimo na tri enake dele in se v zanki vsakokrat odločimo, kako popravimo x_1 oziroma x_2 , da minimum ostaja na intervalu $[x_1, x_2]$. Ko postane interval dovolj majhen, $|x_2 - x_1| < 0.0001$, zaključimo in vrnemo točko $x = (x_1 + x_2)/2$.

Vhod:

Iz standardnega vhoda preberete koordinati y_1 in y_2 .

Izhod:

Na standardni izhod izpišete koordinato x pristanišča.

Primera:

1. Vhod $y_1, y_2 = 3, 7$

Izhod $x = -0.9$

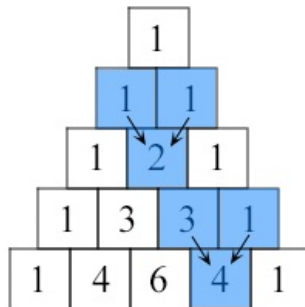
2. Vhod $y_1, y_2 = 3, 6$

Izhod $x = -0.67$

Pascalov trikotnik

Naloga

Program naj izpiše prvih $n + 1$ vrstic Pascalovega trikotnika.



Vhod:

Iz standardnega vhoda preberete število n .

Izhod:

Na standardni izhod izpišete vrstice Pascalovega trikotnika.

Primer:

```
vhod: 7
izhod:
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
[1, 7, 21, 35, 35, 21, 7, 1]
```

Frekvence vodilne številke

Naloga

Sestavi zaporedje naravnih števil z upoštevanjem naslednjih pravil.

1. Izberi poljubno celo število n , večje od 4 ($n > 4$).
2. Če je število sodo, potem je naslednji člen enak $n/2$.
3. Sicer pa je naslednji člen enak $3n + 1$.
4. Zaporedje nadaljujemo toliko časa, dokler ni člen prvič enak 1.

Na koncu sestavimo seznam s frekvencami vodilnih številke v zaporedju.

$f = \{ '1': n_1, '2': n_2, '3': n_3, '4': n_4, '5': n_5, '6': n_6, '7': n_7, '8': n_8, '9': n_9 \}$,

kjer pomenijo vrednosti $n_1 \dots n_9$ število kolikokrat se je pojavila dana vodilna številka v zaporedju.

Vhod:

Iz standardnega vhoda preberete število n .

Izhod:

Na standardni izhod izpišete ustrezen seznam f .

Primer:

```
vhod: 27
izhod: { '1': 31, '2': 18, '3': 14, '4': 16, '5': 7, '6': 6, '7': 7, '8': 6, '9': 7 }
```