

ŠOLSKO TEKMOVANJE PYTHON JAN. 2023



ZVEZA ZA TEHNIČNO KULTURO SLOVENIJE

Skupina Python

Urejevalnik Python se nahaja na strani:

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

- Skupina I 7 razred 4 naloge 100%.
- Skupina J 8 razred 5 nalog 100%.
- Skupina K 9 razred 6 nalog 100%.

1. Uravnovežen 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 2101 v tem sestavu predstavlja število

$$2101_3 = 2 \cdot 3^3 + 1 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0 = 2 \cdot 27 + 9 + 1 = 54 + 9 + 1 = 64_{10}.$$

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 $-$, o , $+$, številске vrednosti pa so enake -1 , 0, 1. Zapis $+oo-oo$ predstavlja število

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

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 $+$, o in $-$.

Izhod:

Na standardni izhod izpišete ustrezno število v desetiškem zapisu.

Primer:

vhod: +oo-oo izhod 234
vhod: -oo+oo izhod -234

2. Uravnovežen trojiški sestav drugič

Naloga

Iščemo pretvorbo v obratni smeri. Za dano naravno število poiščite zapis v uravnoveženem trojiškem sestavu. **Vhod:**

Iz standardnega vhoda preberete naravno število.

Izhod:

Na standardni izhod izpišete ustrezen niz znakov +, o in -, ki predstavlja zapis tega števila v uravnoveženem trojiškem sestavu.

Primer:

vhod: 234 izhod: +oo-oo
vhod: -234 izhod: -oo+oo

3. Faktorizacija

Naloga

Za dano naravno število poiščite praštevila, katerih produkt je to število.

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

Vhod:

Iz standardnega vhoda preberete število.

Izhod:

Na standardni izhod izpišete praštevila, urejena v naraščajočem vrstnem redu, ločena s presledkom, katerih produkt je dano število.

Primeri:

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

4. Podnizi

Naloga:

Dana sta dva niza, prvi je glavni niz, drugi je podniz. Vaša naloga je najti vse indekse mest v glavnem nizu, kjer se začne podniz.

Vhod:

Iz standardnega vhoda preberete niza **glavniniz** in **podniz**, ločimo ju s podpičjem.

Izhod:

Na standardni izhod izpišete vse indekse, ločene s presledkom, kjer se v glavnem nizu pojavi prvi znak podniza.

Primer:

Vhod: na planincah sončece sije na planincah luštno je;je
Izhod: 23 46

5. Frekvence znakov v nizu

Naloga:

Določite frekvence posameznih znakov v nizu.

Vhod:

Iz standardnega vhoda preberete niz **niz**.

Izhod:

Na standardni izhod izpišete frekvence znakov v obliki seznama dvojic, urejenih v slovenskem abecednem vrstnem redu. Seznam naj vsebuje samo znake, ki so se pojavili v nizu.

Primer:

Vhod: na planincah sončece sije na planincah luštno je

Izhod: [(' ', 7), ('a', 6), ('c', 3), ('č', 1), ('e', 4), ('h', 2), ('i', 3), ('j', 2), ('l', 3), ('n', 8), ('o', 2), ('p', 2), ('s', 2), ('š', 1), ('t', 1), ('u', 1)]

6. Obrazila

Naloga:

Vzemimo niz $T = \text{'GATAGACA\$'}$. Zadnji znak '\$' dodamo prvotnemu nizu 'GATAGACA' kot znak, ki zaključuje niz. Njegova vrednost ASCII je najmanjša v nizu.

i	Suffix		i	SA[i]	Suffix
0	GATAGACA\$	Sort →	0	8	\$
1	ATAGACA\$		1	7	A\$
2	TAGACA\$		2	5	ACA\$
3	AGACA\$		3	3	AGACA\$
4	GACA\$		4	1	ATAGACA\$
5	ACA\$		5	6	CA\$
6	CA\$		6	4	GACA\$
7	A\$		7	0	GATAGACA\$
8	\$		8	2	TAGACA\$

Sestavimo seznam obrazil, kot prikazuje slika. Uredimo seznam po abecednem vrstnem redu in določimo permutacijo nizov. Iz slike preberemo, da je ta [8, 7, 5, 3, 1, 6, 4, 0, 2].

Vhod:

Iz standardnega vhoda preberemo niz.

Izhod:

Na izhod izpišemo permutacijo pri urejanju obrazil v abecednem vrstnem redu.

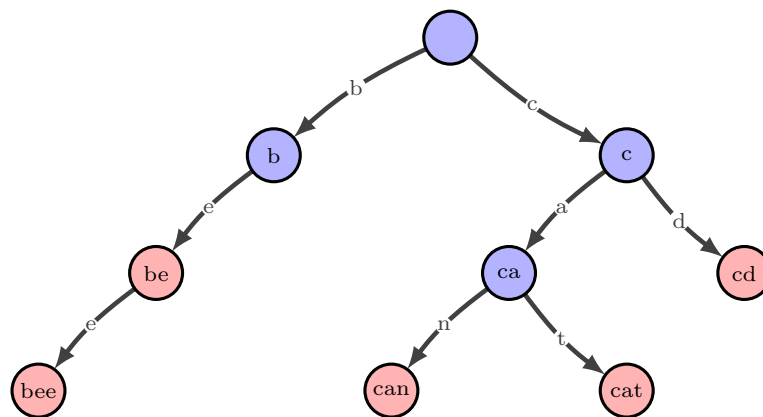
Primeri:

vhod: GATAGACA\$

izhod: [(8, '\$'), (7, 'A\$'), (5, 'ACA\$'), (3, 'AGACA\$'), (1, 'ATAGACA\$'),
(6, 'CA\$'), (4, 'GACA\$'), (0, 'GATAGACA\$'), (2, 'TAGACA\$')]

7. Trie

Spomnimo se aplikacije na mobilnih telefonih, ki nam, ko začnemo tipkati, začne prikazovati mogoča nadaljevanja besede že po nekaj vnesenih znakih. Aplikacija je zelo priročna, ker nam omogoča, da vpisujemo besede, ki so pravilno črkovane. Kako računalnik v vašem prenosnem telefonu lahko poišče mogoča nadaljevanja besed, ki jih vnašate? Slovar v vašem računalniku je zapisan v posebni podatkovni strukturi, ki se imenuje *trie*.



Naloga:

Za dani seznam besed zgradimo drevo *trie*.

Vhod:

Iz standardnega vhoda preberemo seznam besed.

Izhod:

Na standardni izhod izpišemo slovar, ki predstavlja drevo *trie*.

Primeri:

1.

vhod: ['be', 'bee', 'can', 'cat', 'cd']

```
izhod: {"b": {"e": {"$": {}},  
        "e": {"$": {}}},  
       "c": {"a": {"n": {"$": {}},  
                "t": {"$": {}}},  
       "d": {"$": {}}}
```

2.

vhod: ["jablana", "jabolko", "hruška"]

izhod:

```
{"j": {"a": {"b": {"o": {"l": {"k": {"o": {"$": {}}}}}}},
```

```
    "l": {"a": {"n": {"a": {"$: {}}}}}},  
"h": {"r": {"u": {"s": {"k": {"a": {"$: {}}}}}}}
```