

Obvezna naloga

Ideja te naloge je, da jo rešite z eno samo preprosto vrstico v numpyju. Nobenega for ali while, nobenih izpeljanih seznamov ali česa podobnega!

Pravilno napisna funkcija bo dolga eno vrstico, ki bo krajša od vrstice

```
def srecne_drzave(teze, vrednosti, drzave, meja):
```

Tore: napišite funkcijo `srecne_drzave(teze, vrednosti, drzave, meja)`.

- `teze` vsebuje težo uvoženih banan,
- `vrednosti` pove, koliko je posamezna država plačala za te banane,
- `drzave` pa so imena držav. Vse tri tabele so enake dolžine in vsebujejo podatke v enakem vrstnem redu.

Gre, očitno, za to, kar vračata prvi dve funkciji iz prejšnje domače naloge.

Funkcija vrne vrne tabelo z imeni držav, ki so za svoje banane plačale manj od podane meje. Države morajo v tabeli nastopati v enakem vrstnem redu kot v podani tabeli `drzave`. Z drugimi besedami, tabela, ki jo vrnete, mora biti takšna kot tabela `drzave`, vendar le s tistimi državami, ki so plačale manj kot `meja`.

Pomoč: s tabelo nizov lahko delamo enake stvari, kot smo jih na predavanjih s tabelami števil - tudi tabelo držav lahko indeksiramo s tabelami indeksov, s tabelo `True`-jev in `False`-ov ipd.

Podatki iz testov

Semle skopiram podatke iz testov, da lahko spodaj poganjam primere.

```
import numpy as np
```

```
teze = np.array([
    1068570000, 9342850, 612897, 18700200, 49379400, 56, 370860, 43500500, 10144900, 138
    17534600, 244311, 1532300, 196011000, 11942800, 14537700, 126296000, 1109590000, 192
    200695000, 38, 36990300, 1043, 44463300, 2254140, 8458220, 4928940, 34353400, 665230
    6713560,
    2801620, 25146700, 1383, 232316000, 16513200, 15257100, 87426100, 1262980, 984042, 3
    3674190, 1389550, 18517700, 1405, 70, 214838000, 243406, 3811980, 209393, 663050000,
    1023000000, 96376, 42021600, 73296100, 10801, 652604000, 11648, 175679000, 56456100,
    5749150, 496208000, 189785000, 661390000, 306924000, 3990210, 2201580, 252888000, 72
    1035, 77105800, 5007, 20, 6560710, 15098200, 1260, 1308900000, 4985, 16187800, 81313
    27587000, 10744700, 620922, 120931000, 4078, 98410700, 40, 511178, 79888700, 1053610
    21303300, 2243, 6242780, 53954900, 39347600, 58980300, 954379, 45073400, 221254000,
    152588000, 67857700, 2779200, 39630100, 4220, 72308600, 91093500, 505, 423, 682, 11
    1000, 7133440, 192466, 623, 41875700, 20, 564718000, 7, 49, 36951000, 97095, 67025]
)
vrednosti = np.array([
    704414240.0, 5198640.0, 157250.0, 5112190.0, 36917880.0, 630.0, 8000.0, 15652370.0,
```

```

92800.0,
448430.0, 4370.0, 12969130.0, 49730.0, 382490.0, 132893720.0, 11756100.0, 4162040.0,
19999940.0, 114364670.0, 82133940.0, 95349460.0, 290.0, 8672850.0, 1770.0, 34411500.0,
8420020.0, 30988300.0, 246818670.0, 6020.0, 2121188560.0, 1053330.0, 89600.0, 221090.0,
8342040.000000001, 14250450.0, 86454840.0, 1010400.0, 5040470.0, 2960.0, 34260.0, 13
1362790.0,
769760.0, 1500810.0, 310.0, 290.0, 154979800.0, 319310.0, 212700.0, 317670.0, 497390.0,
1794780.0, 762868730.0, 42760.0, 32244710.0, 34409890.0, 8550.0, 704455360.0, 52000.0,
306660.0, 75920.0, 12041100.0, 5823920.0, 354618660.0, 131940180.0, 392402000.0, 176
4178640.0000000005,
331280.0, 202587630.0, 3080.0, 11359250.0, 910640.0, 190.0, 830.0, 67082000.0, 21350.0,
14232640.0, 1920.0, 946332640.0, 24940.0, 11224320.0, 61946980.0, 105558710.0, 33978.0,
55400.0, 100032790.0, 3260.0, 47176630.0, 120.0, 163550.0, 101210240.0, 5584560.0, 2
3255560.0, 4140.0, 7085300.0, 36155240.0, 17821430.0, 54699790.0, 744540.0, 24541610.0,
5114640.0, 158850.0, 102653740.0, 47623360.0, 1565850.0, 25061160.0, 3550.0, 5794539
470.0,
12040.0, 16552220.000000002, 367340.0, 4715430.0, 160.0, 7980060.0, 803750.0, 1060.0,
427873090.0, 130.0, 240.0, 22456700.0, 111600.0, 32400.0]
)
drzave = np.array([
'Russian Federation', 'Botswana', 'Paraguay', 'Tunisia', 'Croatia', 'Cook Islands',
'Egypt, Arab Rep.', 'Bahrain', 'Greenland', 'Vietnam', 'French Polynesia', 'Albania',
'Czech Republic', 'Estonia', 'Trinidad and Tobago', 'Austria', 'Japan', 'Azerbaijan',
'Turkey', 'Vanuatu', 'South Africa', 'Malawi', 'Belarus', 'Mauritania', 'Armenia', '
'China', 'New Caledonia', 'United States', 'Nigeria', 'Burkina Faso', 'Colombia', 'E
'Syrian Arab Republic',
'Senegal', 'Georgia', 'Denmark', 'Bermuda', 'Bahamas, The', 'Dominican Republic', 'V
'Brunei',
'Barbados', 'Maldives', 'Mali', 'Pakistan', 'Cote d'Ivoire', 'Ukraine', 'Andorra', '
'Sweden', 'St. Kitts and Nevis', 'Aruba', 'United Kingdom', 'Kenya', 'Bulgaria', 'Ho
'Belgium', 'Israel', 'Chile', 'Finland', 'Rwanda', 'Bhutan', 'Morocco', 'Iceland', '
'Iran, Islamic Rep.', 'Saudi Arabia', 'Namibia', 'Guatemala', 'Poland', 'Ethiopia(ex
'Occ.Pal.Terr', 'Costa Rica', 'Comoros', 'Lao PDR', 'Hungary', 'Ghana', 'Gambia, The
'Bangladesh', 'Germany', 'Brazil', 'North Macedonia', 'New Zealand', 'Algeria', 'Nep
'Uganda', 'Greece', 'Mozambique', 'Libya', 'Guyana', 'Zimbabwe', 'Switzerland', 'Mol
'Antigua and Barbuda', 'Bolivia', 'Afghanistan', 'Gabon', 'Malta', 'Slovenia', 'Sing
'Lesotho', 'Kazakhstan', 'Netherlands', 'Thailand', 'Kyrgyz Republic', 'Zambia', 'Po
'Indonesia', 'Jordan', 'Angola', 'Ireland', 'Kuwait', 'Micronesia, Fed. Sts.', 'St.
'El Salvador', 'Nicaragua', 'Honduras', 'Burundi', 'Cyprus', 'Australia', 'Cape Ver
'Serbia, FR(Serbia/Montenegro)', 'Palau', 'France', 'Other Asia, nes', 'Tonga', 'Bos
'Grenada', 'Sri Lanka']
)

```

Rešitev

Cene banan dobimo tako, da delimo vrednosti s težami.

vrednosti / teze

```
array([ 0.65921207,  0.55642978,  0.2565684 ,  0.27337622,  0.74763727,
        11.25      ,  0.02157148,  0.35982046,  0.51821112,  0.66646091,
        2.1269281 ,  0.80001213,  5.87365591,  0.73963079,  0.20355203,
        0.24961822,  0.67799113,  0.98436715,  0.28629288,  1.01617391,
        0.76671445,  1.03940068,  0.32573804,  1.04648286,  0.47509634,
        7.63157895,  0.23446282,  1.6970278 ,  0.77393041,  0.10438127,
        1.39987137,  1.70828211,  0.90204463,  0.37102757,  6.53637351,
        0.48635274,  0.15689589,  0.0319815 ,  0.08792008,  1.00506146,
        0.38471621,  0.50517404,  0.93402088,  0.9888905 ,  0.80001267,
        5.12221023,  0.8008658 ,  6.85062987,  0.58193884,  0.78973109,
        0.37090896,  0.55396351,  0.08104732,  0.22064057,  4.14285714,
        0.72137983,  1.31184112,  0.05579777,  1.51709943,  0.75015584,
        0.97827988,  0.47258725,  0.64652061,  0.74571723,  0.44367892,
        0.76733656,  0.46946413,  0.79159337,  1.07945302,  4.46428571,
        0.26784562,  1.25730523,  0.0945195 ,  0.17564641,  0.42914268,
        1.0130054 ,  0.71465728,  0.69520868,  0.59329896,  0.5737697 ,
        1.04722308,  0.15047375,  0.80109626,  4.27184466,  0.78972525,
        0.03684206,  2.375      ,  0.80193237,  0.8699994 ,  4.26403036,
        17.        ,  0.58057893,  0.94267131,  1.52380952,  0.72299843,
        5.00300903,  0.69338144,  0.76182616,  0.50783805,  0.02795784,
        0.5873792 ,  0.47179633,  0.08922216,  0.82718898,  0.79941148,
        0.47938517,  3.        ,  0.31994726,  1.26689056,  0.53004053,
        0.62252241,  0.64973714,  0.18518519,  0.15281952,  1.84574231,
        1.1349591 ,  0.67010114,  0.45292292,  0.92742475,  0.78013033,
        0.544481   ,  0.84379695,  0.1651268 ,  0.56914515,  0.78974058,
        0.67275107,  0.70181217,  0.56341753,  0.63237691,  0.84123223,
        0.80136236,  0.62630857,  0.79207921,  1.11111111,  17.65395894,
        0.1479105 ,  0.04403279,  0.08442285,  0.16        ,  1.11868327,
        4.17606227,  1.70144462,  0.73092295,  1.        ,  0.75767567,
        18.57142857,  4.89795918,  0.60774269,  1.14938977,  0.48340172])
```

Zanimajo nas elementi, ki so manjši od določene meje. Naj bo

`meja = 0.4`

Potem imamo

`vrednosti / teze < meja`

```
array([False, False,  True,  True, False, False,  True,  True, False,
        False, False, False, False, False,  True,  True, False, False,
        True, False, False, False,  True, False, False, False,  True,
        False, False,  True, False, False, False,  True, False, False,
        True,  True,  True, False,  True, False, False, False, False,
        False, False, False, False, False,  True, False,  True,  True,
        False, False, False,  True, False, False, False, False, False,
        False, False, False, False, False, False, False,  True, False,
```

```

True, True, False, False, False, False, False, False, False,
True, False, False, False, True, False, False, False, False,
False, False, False, False, False, False, False, False, False,
True, False, False, True, False, False, False, False, True,
False, False, False, False, True, True, False, False, False,
False, False, False, False, False, True, False, False, False,
False, False, False, False, False, False, False, False, False,
True, True, True, True, False, False, False, False, False,
False, False, False, False, False, False, False, False, False,

```

Če želimo iz seznama držav dobiti tiste, pri katerih je v tem seznamu True, napišemo

```

drzave[vrednosti / teze < meja]

array(['Paraguay', 'Tunisia', 'Benin', 'Uruguay', 'Lebanon', 'Malaysia',
      'Trinidad and Tobago', 'Argentina', 'South Africa', 'Mauritania',
      'China', 'Nigeria', 'Burkina Faso', 'Colombia',
      'Syrian Arab Republic', 'Barbados', 'Mali', 'Pakistan', 'Niger',
      'Chile', 'Rwanda', 'Bhutan', 'Guatemala', 'Costa Rica', 'Nepal',
      'Uganda', 'Zimbabwe', 'Bolivia', 'Afghanistan', 'Thailand',
      'El Salvador', 'Nicaragua', 'Honduras', 'Burundi'], dtype='<U29')

```

Naša funkcija je potemtakem:

```

def srecne_drzave(teze, vrednosti, drzave, meja):
    return drzave[vrednosti / teze < meja]

```

Dodatna naloga

Ta je tako, za zraven. Neobvezno.

Dam vam funkcijo

```

def indeks_drzave(drzave, drzava):
    return np.where(drzave == drzava)[0][0]

```

ki vrne indeks države v tabeli držav.

Napišite funkcijo `vec_kot(teze, drzave, drzava)`, ki prejme tabelo s težami, tabelo z državami in ime neke države. Vrne število držav, ki so kupile več banan kot podana država.

Rešitev

`indeks_drzave` potrebujemo samo zato, da izvemo, koliko banan je kupila ta država.

```
drzava = "Slovenia"
```

```
indeks_drzave(drzave, "Slovenia")
```

```
np.int64(116)
```

```
teze[indeks_drzave(drzave, "Slovenia")]
```

```
np.int64(53954900)
```

Ko vemo, koliko kilogramov banan je kupila Slovenija, lahko sestavimo tabelo, ki bo vsebovala True za vse teže, ki so večje od te.

```
teze > teze[indeks_drzave(drzave, "Slovenia")]
```

```
array([ True, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False,  True, False,
        False,  True,  True, False,  True,  True,  True, False, False,
        False, False, False, False, False, False, False,  True, False,  True,
        False, False, False, False,  True, False, False,  True, False,
        False, False, False, False, False, False, False, False, False, False,
        False,  True, False, False, False,  True,  True, False, False,
        True, False, False,  True, False,  True, False,  True,  True,
        False, False, False, False,  True,  True,  True,  True, False,
        False,  True, False, False, False, False, False,  True, False,
        False, False, False, False,  True, False, False,  True,  True,
        False, False, False, False,  True, False,  True, False, False,
        True, False,  True, False, False, False, False, False, False,
        False,  True, False, False,  True, False, False, False,  True,
        True, False, False, False,  True,  True, False, False, False,
        True, False,  True, False, False, False, False, False, False,
        True, False, False, False, False, False, False])
```

Da ugotovimo, koliko je True-jev, seštejemo elemente te tabele.

```
sum(teze > teze[indeks_drzave(drzave, "Slovenia")])
```

```
np.int64(41)
```

In funkcija je:

```
def vec_kot(teze, drzave, drzava):
    return sum(teze > teze[indeks_drzave(drzave, drzava)])
```