Home (/) / Book (/book/)
/ Chapter 2: The Core Python Language I (/book/chapter-2-the-core-python-language-i/)
/ Problems (/book/chapter-2-the-core-python-language-i/questions/problems/)
/ P2.6 (/book/chapter-2-the-core-python-language-i/questions/problems/p26/)
/ Earth Similarity Index

Earth Similarity Index

Question P2.6.3

The *Earth Similarity Index* (ESI) attempts to quantify the physical similarity between an astronomical body (usually a planet or moon) and Earth. It is defined by

$$\mathrm{ESI}_j = \prod_{i=1}^{n}\left(1 - \left|\frac{x_{i,j} - x_{i,\oplus}}{x_{i,j} + x_{i,\oplus}}\right|\right)^{w_i/n}$$

where the parameters $x_{i,j}$ are described, and their terrestrial values, $x_{i,\oplus}$ and weights, w_i given in the table below. The radius, density and escape velocities are taken *relative to* the terrestrial values. The ESI lies between 0 and 1, with the values closer to 1 indicating closer similarity to Earth (which has an ESI of exactly 1: Earth is identical to itself!)

i	**Parameter**	**Earth Value,** $x_{i,\oplus}$	**Weight,** w_i
1	Radius	1.0	0.57
2	Density	1.0	1.07
3	Escape velocity, v_{esc}	1.0	0.7
4	Surface temperature	288 K	5.58

The file ex2-6-g-esi-data.txt (/static/media/problems/P2.6/ex2-6-g-esi-data.txt) contains the above parameters for a range of astronomical bodies. Use these data to calculate the ESI for each of the bodies. Which has properties "closest" to those of the Earth?

Solution P2.6.3

The provided data file consists of three header lines followed by the data in 9 columns. The name of the astronomical body is in the first column, and the required parameters are in the third, fourth, sixth and eighth columns. We can't simply split each entire row on whitespace because some of the names contain spaces, so we extract the name from the first 15 characters and split the remaining row on its whitespace. This places the parameters in fields indexed at 1, 2, 4 and 6 (starting at 0).

The following program defines a tuple of field indexes and extracts the parameters for each row using it. For each row (representing an astronomical body), the ESI is calculated (using the hard-coded terrestrial values) as a product over these parameters and output.

```
f = open('ex2-6-g-esi-data.txt', 'r')
# skip the three header lines
f.readline()
f.readline()
f.readline()

# The column indexes of fields *after the first, name field* within the
# provided data table for the properties needed to calculate the ESI
cols = (1, 2, 4, 6)
n = len(cols)
# The terrestrial values of those parameters and their weights
x_earth, w = (1, 1, 1, 288.), (0.57, 1.07, 0.70, 5.58)

print('-'*29)
print('Planet/Satellite Name   ESI  ')
print('-'*29)
for line in f.readlines():
    name = line[:15].lstrip()
    fields = line[15:].split()
    ESI = 1
    for i, col in enumerate(cols):
        xi = float(fields[col])
        ESI *= (1 - abs((xi - x_earth[i])/(xi + x_earth[i])))**w[i]
    ESI = ESI**(1/n)
    print('{:<21s}   {:5.3f}'.format(name, ESI))
print('-'*29)
```

The output from this program is given below.

```
-----------------------------
Planet/Satellite Name   ESI
-----------------------------
Earth                   1.000
Mars                    0.697
Mercury                 0.595
Moon                    0.557
Venus                   0.444
Io                      0.363
Jupiter                 0.293
Titan                   0.243
GJ 581 g                0.890
GJ 581 b                0.483
HD 96167 b              0.465
WASP-26 b               0.094
-----------------------------
```