

Inflation

Problem ID: p07inflation

This project is an exercise in file processing, lists and tuples. Write a program that reads information about consumer price index from an input file and prints out various information about the data. The data in the input files are obtained from the website of Statistics Iceland, <https://statice.is/>. A typical file (`data9091.txt`), which contains data for the years 1990 and 1991, looks like this:

```
1990M01 139.3
1990M02 141.5
1990M03 142.7
1990M04 143.1
1990M05 144.4
1990M06 145.4
1990M07 146.4
1990M08 146.8
1990M09 146.8
1990M10 147.2
1990M11 148.2
1990M12 148.6
1991M01 149.5
1991M02 150.0
1991M03 150.3
1991M04 151.0
1991M05 152.8
1991M06 154.9
1991M07 156.0
1991M08 157.2
1991M09 158.1
1991M10 159.3
1991M11 160.0
1991M12 159.8
```

The first column contains year/month in ascending order. The second column contains the index for the corresponding year/month. The data in an input file **SHOULD** be read into a list of tuples.

Input

The input is a name of a file to be analyzed, e.g. `data9091.txt` or any other file name ending with `.txt`. Each line in an input file contains a string denoting a year/month and a float denoting the index for the corresponding year/month. These two fields are separated by a whitespace.

Output

When the input file cannot be opened, no output is generated. Otherwise, the output consists of the following information:

1. One tuple (y, i) in line for each line in the input file, where y is year/month (a string) and i is the corresponding index (a float).
2. One tuple (y, f, l) in line for each year in the input file, where y is a year (an int) and f and l are the first and the last indices, respectively (both values are floats), for year y .
3. One tuple (y, i) in line for each year in the input file, where y is a year (an int) and i is the calculated inflation (a float), rounded to two decimal digits, given the first and the last index for year y .

Sample Input 1

data08.txt

Sample Output 1

('2008M01', 282.3)
('2008M02', 286.2)
('2008M03', 290.4)
('2008M04', 300.3)
('2008M05', 304.4)
('2008M06', 307.1)
('2008M07', 310.0)
('2008M08', 312.8)
('2008M09', 315.5)
('2008M10', 322.3)
('2008M11', 327.9)
('2008M12', 332.9)
(2008, 282.3, 332.9)
(2008, 17.92)

Sample Input 2

data9091.txt

Sample Output 2

('1990M01', 139.3)
('1990M02', 141.5)
('1990M03', 142.7)
('1990M04', 143.1)
('1990M05', 144.4)
('1990M06', 145.4)
('1990M07', 146.4)
('1990M08', 146.8)
('1990M09', 146.8)
('1990M10', 147.2)
('1990M11', 148.2)
('1990M12', 148.6)
('1991M01', 149.5)
('1991M02', 150.0)
('1991M03', 150.3)
('1991M04', 151.0)
('1991M05', 152.8)
('1991M06', 154.9)
('1991M07', 156.0)
('1991M08', 157.2)
('1991M09', 158.1)
('1991M10', 159.3)
('1991M11', 160.0)
('1991M12', 159.8)
(1990, 139.3, 148.6)
(1991, 149.5, 159.8)
(1990, 6.68)
(1991, 6.89)

Sample Input 3

data2022.txt

Sample Output 3

```
('2020M01', 469.8)
('2020M02', 474.1)
('2020M03', 475.2)
('2020M04', 477.5)
('2020M05', 480.1)
('2020M06', 482.2)
('2020M07', 482.9)
('2020M08', 485.1)
('2020M09', 487.0)
('2020M10', 489.1)
('2020M11', 489.3)
('2020M12', 490.3)
('2021M01', 490.0)
('2021M02', 493.4)
('2021M03', 495.8)
('2021M04', 499.3)
('2021M05', 501.4)
('2021M06', 502.7)
('2021M07', 503.5)
('2021M08', 505.8)
('2021M09', 508.2)
('2021M10', 511.2)
('2021M11', 513.0)
('2021M12', 515.3)
('2022M01', 517.9)
('2022M02', 523.9)
('2022M03', 528.8)
('2022M04', 535.4)
('2022M05', 539.5)
('2022M06', 547.1)
('2022M07', 553.5)
('2022M08', 555.1)
('2022M09', 555.6)
('2022M10', 559.3)
('2022M11', 560.9)
('2022M12', 564.6)
('2023M01', 569.4)
('2023M02', 577.3)
('2023M03', 580.7)
('2023M04', 588.3)
('2023M05', 590.6)
('2023M06', 595.6)
('2023M07', 595.8)
('2023M08', 597.8)
('2023M09', 599.9)
(2020, 469.8, 490.3)
(2021, 490.0, 515.3)
(2022, 517.9, 564.6)
(2023, 569.4, 599.9)
(2020, 4.36)
(2021, 5.16)
(2022, 9.02)
(2023, 5.36)
```