



Laboratorio di Elementi di Bioinformatica

Laurea Triennale in Informatica
(codice: E3101Q116)

AA 2018/2019

Esercizio (18 ottobre 2018)

Docente: Raffaella Rizzi

[Esercizio]

Produrre uno script Python che legga un file (da specificare come argomento della riga di comando) contenente le precipitazioni piovose mensili (in mm) in un certo luogo e in un periodo di N anni (fare riferimento al file `input-precipitazioni.txt`) e produrre in standard output:

- le precipitazioni medie mensili (lungo gli anni)
- per ogni anno le precipitazioni totali
- per ogni anno il numero di mesi con precipitazione di almeno 100 mm

Per il formato di output fare riferimento al file `output-precipitazioni.txt`

[Esercizio]

Produrre uno script Python che legga un file (da specificare come argomento della riga di comando) contenente le precipitazioni piovose mensili (in mm) in un certo luogo e in un periodo di N anni (fare riferimento al file `input-precipitazioni.txt`) e produrre in standard output:

- le precipitazioni medie mensili (lungo gli anni)
- per ogni anno le precipitazioni totali
- per ogni anno il numero di mesi con precipitazione di almeno 100 mm

Per il formato di output
`precipitazioni.`

I nomi dei mesi sono da indicare solo con le prime tre lettere del mese in maiuscolo (es. Gennaio → GEN)

[Esercizio]

Produrre uno script Python che legga un file (da specificare come argomento della riga di comando) contenente le precipitazioni piovose mensili (in mm) in un certo luogo e in un periodo di N anni (fare riferimento al file `input-precipitazioni.txt`) e produrre in standard output:

- le precipitazioni medie mensili (lungo gli anni)
- per ogni anno le precipitazioni totali
- per ogni anno il numero di mesi con precipitazione di almeno 100 mm

Per il formato di output
precipitazioni

I valori decimali devono essere prodotti in output con una sola cifra decimale

[Suggerimenti...

Per produrre in output un decimale con una sola cifra decimale usare l'operatore di formato %

```
f = 45.676455  
'%.1f' % f          # '45.7'
```

[Suggerimenti...

La funzione `int(value)` converte in intero il valore passato come argomento.

```
int('13')      # 13  
int(13.5)      # 13
```

[Suggerimenti...

La funzione `float(value)` converte in decimale il valore passato come argomento.

```
float('13.5')    # 13.5  
float(13)         # 13.0
```

[Suggerimenti...

La funzione `sum(arg)` può prendere come argomento una lista o una tupla e restituisce la somma degli elementi

```
sum([1,2,3,4,5])      # 15
```


[Suggerimenti...

Il metodo `upper()` delle stringhe restituisce una copia in maiuscolo della stringa invocante

```
'ciao'.upper()      # 'CIAO'
```

[Suggerimenti...

Il metodo `rstrip(arg_str)` delle stringhe restituisce una copia della stringa invocante dopo avere rimosso il più lungo suffisso contenente solo caratteri della stringa *arg_str* passata come argomento. Se l'argomento viene omesso, allora il metodo rimuove tutti i caratteri di tipo spazio (tabulazione, *newline*, etc.))

```
'aaaa' .rstrip(' ')      # 'aaaa'
'aaaa' .rstrip()         # 'aaaa'
'aaaa\n'.rstrip()       # 'aaaa'
```

[Suggerimenti...

Il metodo `rstrip(arg_str)` delle stringhe restituisce una copia della stringa invocante dopo avere rimosso il più lungo suffisso contenente solo caratteri della stringa `arg_str` passata come argomento. Se l'argomento viene omesso, allora il metodo rimuove tutti i caratteri di tipo spazio (tabulazione, *newline*, etc.)

```
'aaaa' .rstrip(' ')      # 'aaaa'
'aaaa' .rstrip()         # 'aaaa'
'aaaa\n'.rstrip()       # 'aaaa'
```

Il metodo `lstrip()` rimuove un analogo prefisso e `strip()` rimuove sia un prefisso che un suffisso

[Suggerimenti...

Il metodo `split(delim)` delle stringhe restituisce la lista degli elementi ottenuti separando la stringa invocante tramite il delimitatore passato come argomento. Se l'argomento viene omesso la stringa invocante viene separata usando spazi di qualsiasi tipo.

```
'aa aa aa'.split(' ')      # ['aa', 'aa', 'aa']  
'aa aa aa'.split()        # ['aa', 'aa', 'aa']  
'aa  aa      aa'.split()  # ['aa', 'aa', 'aa']  
'aa\taa\taa'.split()      # ['aa', 'aa', 'aa']
```

[Suggerimenti...

Il metodo `pop(index)` delle liste rimuove e restituisce l'elemento di indice *index* della lista invocante.

```
[1,2,3,4].pop(1)      # 2
```

[Suggerimenti...

Il metodo `append(value)` delle liste aggiunge in coda alla lista invocante l'argomento *value*.

```
[1,2,3,4].append(5)      # [1,2,3,4,5]
```

[Suggerimenti...

Il metodo `count (value)` delle sequenze restituisce il numero di elementi della sequenza invocante che sono uguali all'argomento *value*

```
[True,False,True,True].count(True)    # 3  
'aabbabbbba'.count('a')              # 4
```

[Suggerimenti...

```
[expression for elem in sequence]
```

Restituisce una lista di valori, dove ciascun valore (nell'ordine) è dato dalla valutazione dell'espressione *expression* iterando per ciascun elemento nella sequenza *sequence*

```
l = [s+'l' for s in 'ciao']  
    # ['cl', 'il', 'al', 'ol']
```


[Suggerimenti...

With statement per aprire un file in lettura e leggere le sue righe

```
with open(file_name, 'r') as input_file:  
    list_of_rows = input_file.readlines()
```