TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 09 (30 minutos) Turno 2ª feira 09:00-10:30
Nome:
Número:
Data:
Curso:

Capítulo 8 - Dicionários

Escreva uma função em Python que recebe um dicionário cujos valores associados às chaves correspondem a listas de inteiros e que devolve o dicionário que se obtém "invertendo" o dicionário recebido, no qual as chaves são os inteiros que correspondem aos valores do dicionário original e os valores são as chaves do dicionário original às quais os valores estão associados. Por exemplo:

```
>>> inverte_dic({'a': [1, 2], 'b': [1, 5], 'c': [9], 'd': [4]})
{1: ['a', 'b'], 2: ['a'], 4: ['d'], 5: ['b'], 9: ['c']}
```

Solução:

```
def inverte_dic(d):
  res = {}
  for e in d:
    for v in d[e]:
      if v in res:
        res[v] = res[v] + [e]
      else:
        res[v] = [e]
  return res
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 09 (30 minutos) Turno 2ª feira 10:30-12:00
Nome:
Número:
Data:
Curso:

Capítulo 8 - Dicionários

Escreva a função `media_chave` que recebe um dicionário (cujos valores associados às chaves são listas não vazias contendo números inteiros) e uma chave e devolve a média dos valores associados a essa chave. Por exemplo,

```
>>> d1 = {'a' : [1, 2], 'b' : [3, 4]}
>>> media_chave(d1, 'a')
1.5
```

Solução:

```
def media_chave(d, c):
    if c in d:
        soma = 0
        for n in d[c]:
            soma = soma + n
        return soma / len(d[c])
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 09 (30 minutos) Turno 4ª feira 8:00-9:30
Nome:
Número:
Data:
Curso:

Capítulo 8 - Dicionários

Escreva a função `multiplos_filtrados` que recebe um natural `n` e um predicado `pred` e que devolve o dicionário cujas chaves são os naturais inferiores ou iguais a `n` que satisfazem o predicado `pred` e cujos valores associados às chaves são tuplos com os `n` primeiros múltiplos da `chave`. Por exemplo,

```
>>> multiplos_filtrados(4, lambda x: x % 2 == 0)
{2: [2, 4, 6, 8], 4: [4, 8, 12, 16]}
```

Solução:

```
def multiplos_filtrados(n, pred):
    res = {}
    for i in range (1, n+1):
        if pred(i):
            t = ()
            for j in range(1, n+1):
                t = t + (i * j,)
            res[i] = t

    return res
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 09 (30 minutos) Turno 5ª feira 10:30-12:00
Nome:
Número:
Data:
Curso:

Capítulo 8 - Dicionários

Escreva a função `maior_valor` que recebe um dicionário, cujos valores associados às chaves podem ser inteiros positivos ou listas (não vazias) de inteiros positivos, e que devolve o maior inteiro existente como valor no dicionário. Por exemplo,

```
>>> maior_valor({'a' : 3, 'b' : [6, 3, 90], 'c' : 20, \
                 'd' : [1, 33, 12]})
90
```

Solução 1:

```
def maior_valor(d):
    maior = 0
    for k in d: # calcula-se o maior
        if isinstance(d[k], int): # it's an int
            if d[k] > maior:
                maior = d[k]
        else:                     # it's a list of int
            for e in d[k]:
                if e > maior:
                    maior = e
     return maior
```

Solução 2:

```
def maior_valor(d):
    lvals = []
    for k in d: # calcula-se o maior
        if isinstance(d[k], int): # it's an int
            lvals += [ d[k] ]
        else:                     # it's a list of int
            for e in d[k]:
                lvals += [ e ]
     return max(lvals)
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 09 (30 minutos) Turno 6ª feira 12:30-14:00
Nome:
Número:
Data:
Curso:

Capítulo 8 - Dicionários

Escreva a função `junta` que recebe dois dicionários, cujos valores associados às chaves correspondem a listas, e devolve o dicionário que contém todas as chaves contidas em pelo menos um dos dicionários e o valor associado a cada chave corresponde à lista obtida pela "união" (no sentido de conjuntos) das listas correspondendo às chaves existentes nos dicionários. Por exemplo,

```
>>> d1 = {'a' : [1, 2], 'b' : [3, 4]}
>>> d2 = {'b' : [4, 5], 'c' : [6, 7]}
>>> junta(d1, d2)
{'a': [1, 2], 'b': [3, 4, 5], 'c': [6, 7]}
```

Solução:

```
def junta(d1, d2):
    for c in d2:
       if c in d1:
            for el in d2[c]:
               if el not in d1[c]:
                    d1[c] = d1[c] + [el]
       else:
            d1[c] = d2[c]
     return d1
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 09 (30 minutos) Turno 6ª feira 14:00-15:30
Nome:
Número:
Data:
Curso:

Capítulo 8 - Dicionários

Escreva a função `soma_valores` que recebe um dicionário cujos valores associados às chaves são ou inteiros ou listas (não vazias) de inteiros e que devolve a soma de todos os valores existentes no dicionário. Por exemplo,

```
>>> soma_valores({'a' : 3, 'b' : [6, 3, 90], 'c' : 20, \
                  'd' : [0, 33, 12]})
167
```

Solução:

```
def soma_valores(d):
    soma = 0
    for c in d:
        if isinstance(d[c], int):
            soma = soma + d[c]
        else:
            for e in d[c]:
                soma = soma + e
    return soma
```

TÉCNICO LISBOA

Fundamentos de Programação - 2018/2019 Aula Prática 09 (30 minutos) Turno 6ª feira 15:30-17:00
Nome:
Número:
Data:
Curso:

Capítulo 8 - Dicionários

Suponha que os preços unitários dos produtos vendidos por uma loja são representados por um dicionário cujas chaves são os nomes dos produtos e os valores associados às chaves correspondem a uma lista com o preço unitário e o valor do IVA associado ao produto. Escreva uma função que recebe a lista dos preços unitários e as compras de um cliente (representada por um dicionário cujas chaves são os nomes dos produtos comprados e o valor associado à chave representa a quantidade desse produto comprada) e que calcula o valor a pagar. Por exemplo,

```
>>> p_u = {'choc' : [2.5, 0.23], 'sumo' : [1.25, 0.12], \
                'pizza' : [4.5, 0.12]}

>>> compras = {'choc' : 3, 'pizza' : 2}
>>> valor_a_pagar(p_u, compras)
    19.305
```

Solução:

```
def valor_a_pagar(precos_unit, compras):
    conta = 0
    for prod in compras:
        conta += (precos_unit[prod][0] * compras[prod]) * \
                 (1 + precos_unit[prod][1])
    return conta
```