

Soal Unguided Rabu, 10 Mei 2023

Praktikum Algoritma dan Pemrograman

01. Filter Duplikasi

Buatlah fungsi **filter_duplikasi(list, batas)** yang dapat menghapus seluruh bilangan yang terduplikasi dalam jumlah yang melebihi **batas** yang ditentukan. **List** berisi bilangan bulat yang tidak terurut dan ada kemungkinan terduplikasi. Sedangkan **batas** adalah bilangan bulat yang menyatakan jumlah duplikasi maksimal yang diperbolehkan. Jika ada satu atau beberapa bilangan di dalam list yang sama (duplikat) dalam jumlah yang melebihi batas, maka kelebihanannya akan dihapus dari list tersebut.

Silahkan perhatikan contoh yang diberikan.

Contoh:

`filter_duplikasi([1, 2, 4, 2, 1, 2, 3, 2], 2)`, berarti:

- `list = [1, 2, 4, 2, 1, 2, 3, 2]`
- `batas = 2`

Berarti didapatkan:

- Angka 1 ada 2 di dalam list (masih sesuai batas)
- Angka 2 ada 4 di dalam list (sudah melebihi batas, kelebihan 2)
- Angka 3 ada 1 di dalam list (masih di bawah batas)
- Angka 4 ada 1 di dalam list (masih di bawah batas)

Maka angka 2 yang melebihi batas dibuang, sehingga menghasilkan list = **[1, 2, 4, 2, 1, 3]**

Perhatikan:

- Fungsi anda langsung cetak output, tidak perlu return.
- List dipastikan tidak akan kosong
- $0 < \text{batas} < 100$
- Angka yang dihapus adalah angka yang di sebelah kanan (jika jumlah duplikatnya melebihi batas).

For example:

Test	Result
<code>filter_duplikasi([1, 2, 4, 2, 1, 2, 3, 2], 2)</code>	<code>[1, 2, 4, 2, 1, 3]</code>
<code>filter_duplikasi([20,37,20,21], 1)</code>	<code>[20, 37, 21]</code>

02. Nol Harus di Depan

Buatlah fungsi **nol_depan(list)** yang akan memindahkan seluruh elemen bernilai **0** ke **depan** (sebelah kiri) dari list tersebut. Elemen yang nilainya bukan 0 tidak perlu diubah urutannya. Silahkan perhatikan contoh yang diberikan.

Contoh:

list = **[1, 2, 3, 0, 4, 0]**, setelah seluruh elemen bernilai 0 dipindahkan ke depan menjadi **[0, 0, 1, 2, 3, 4]**.

list = **[0, 7, 0, 2, 0, 1, 0, 4, 0, 3, 2]**, maka akan menghasilkan **[0, 0, 0, 0, 0, 7, 2, 1, 4, 3, 2]**

Perhatikan:

- len(list) minimal 1
- Lakukan print output di dalam fungsi, sehingga fungsi anda tidak perlu return.

For example:

Test	Result
nol_depan([1, 2, 3, 0, 4, 0])	[0, 0, 1, 2, 3, 4]
nol_depan([0, 7, 0, 2, 0, 1, 0, 4, 0, 3, 2])	[0, 0, 0, 0, 0, 7, 2, 1, 4, 3, 2]

03. Pencarian Operator

Buatlah fungsi **cari_berdasar_operator(contacts, operator)** yang dapat menampilkan nama-nama orang yang menggunakan no hp dari operator yang diberikan.

Variabel contacts adalah sebuah dictionary, contohnya sebagai berikut:

```
contacts = {  
  
'Badu' : '089879938817',  
  
'Hana' : '0818662514110',  
  
'Seto' : '08122290909',  
  
'Adi' : '0856808080012',  
  
'Pace' : '0858000000000'  
}
```

Sedangkan variabel operator adalah sebuah string yang kemungkinan nilainya adalah:

- telkomsel (nomor hp diawali 0812 atau 0822)
- im3 (nomor hp diawali dengan 0856 atau 0857 atau 0858)
- three (nomor hp diawali dengan 0898)
- xl (nomor hp diawali dengan 0817 atau 0818)

Contoh

Pemanggilan

```
contacts = {  
  
'Badu' : '089879938817',  
  
'Hana' : '0818662514110',  
  
'Seto' : '08122290909',  
  
'Adi' : '0856808080012',  
  
'Pace' : '0858000000000'  
}  
  
operator = 'im3'  
  
hasil = cari_berdasar_operator(contacts, operator)  
  
print(hasil)
```

Output

```
['Adi', 'Pace']
```

Penjelasan

Operator yang diminta adalah im3, sehingga yang menggunakan im3 adalah Adi dan Pace (diurutkan berdasarkan nama)

Batasan

- Fungsi anda harus return sebuah list terurut berisi nama-nama yang nomor hp-nya sesuai dengan operator yang diberikan.
- Operator dipastikan hanya 4 kemungkinan (telkomsel, im3, three dan xl)
- Contacts minimal berisi 1 data.

For example:

Test	Result
<pre>contacts = { 'Badu' : '089879938817', 'Hana' : '0818662514110', 'Seto' : '08122290909', 'Adi' : '0856808080012', 'Pace' : '0858000000000' } operator = 'im3' hasil = cari_berdasar_operator(contacts, operator) print(hasil)</pre>	<pre>['Adi', 'Pace']</pre>

04. Hitung Nilai Akhir Berdasarkan n Nilai Terbaik

Buatlah fungsi `hitung_nilai_akhir(daftar_nilai, n)` yang dapat digunakan untuk menghitung nilai akhir berdasarkan rata-rata dari n nilai terbaik. Parameter `daftar_nilai` adalah sebuah dictionary dengan format seperti contoh berikut ini:

```
daftar_nilai = {  
    'Udin' : [65, 74, 56, 80, 82, 94],  
    'Atun' : [98, 84, 82, 88],  
    'Tejo' : [85, 86]  
}
```

Udin memiliki 6 nilai yaitu 65, 74, 56, 80, 82 dan 94.

Atun memiliki 4 nilai yaitu 98, 84, 82, 88

Tejo memiliki 2 nilai, yaitu 85, 86

Contoh

Pemanggilan

```
daftar_nilai = {  
    'Udin' : [65, 74, 56, 80, 82, 94],  
    'Atun' : [98, 84, 82, 88],  
    'Tejo' : [85, 86]  
}
```

`n = 2`

`hitung_nilai_akhir(daftar_nilai, n)`

Output

Atun 93.000000

Tejo 85.500000

Udin 88.000000

Penjelasan

`n = 2`, sehingga nilai akhir akan dihitung dari **2 nilai terbaik**.

Udin memiliki 6 nilai yaitu 65, 74, 56, 80, **82** dan **94**. Rata-rata = $(94 + 82) / 2 = 88.000000$ (dalam format float %f)

Atun memiliki 4 nilai yaitu **98**, 84, 82, **88**. Rata-rata = $(98 + 88) / 2 = 93.000000$ (dalam format float %f)

Tejo memiliki 2 nilai, yaitu **85**, **86**. Rata-rata = $(85+86)/2 = 85.500000$ (dalam format float %f)

Kemudian outputnya diurutkan berdasarkan nama, sehingga urutan outputnya adalah Atun, Tejo dan Udin.

Batasan

- Fungsi anda tidak perlu return, lakukan `print()` di dalam fungsi.

- Dictionary daftar nilai minimal memuat 1 nama, kemudian jumlah nilai setiap nama minimal akan sama dengan n.
- Untuk menampilkan rata-rata gunakan format %f (float) supaya hasilnya seperti yang dicontohkan.

For example:

Test	Result
<pre>daftar_nilai = { 'Udin' : [65, 74, 56, 80, 82, 94], 'Atun' : [98, 84, 82, 88], 'Tejo' : [85, 86] } n = 2 hitung_nilai_akhir(daftar_nilai, n)</pre>	<pre>Atun 93.000000 Tejo 85.500000 Udin 88.000000</pre>