

Praktikum PAP 2 - Fungsi

Tujuan	<ol style="list-style-type: none"> 1. Mampu membuat dekomposisi dari abstraksi suatu masalah dunia nyata menjadi fungsi atau sub-program 2. Mampu mengumpulkan fungsi menjadi file pustaka dan membuat file driver untuk memeriksa file pustaka (2 program)
Ketentuan	<ol style="list-style-type: none"> 1. Praktikan tidak boleh menggunakan peralatan selain yang terdapat di lab, kecuali atas ijin instruktur 2. Dalam mengerjakan problem set, tidak boleh menggunakan built-in function, kecuali yang di ijin dalam problem statement.

Semua fungsi yang dibuat disimpan dalam file **pap2lib.py** dan akan dipakai/inklusikan dalam file program utama **pap2drv.py** dengan pemanggilan **from pap2lib import *** diawal program utama.

A. Konversi

1. Buatlah fungsi **cmToM(c)** untuk merubah masukan **c** (dalam centimeter) menjadi **m** dan **c** (meter, centimeter), contoh masukan = 120 menjadi (1,20) yang artinya 1 m 20 cm, **c=95** menjadi (0,95) artinya 0 m 95 cm, dimana output fungsi adalah tuple. Rumus $m = c/100$, $cm = c \% 100$.
2. Modifikasi fungsi diatas menjadi **hariToTahun(hari)**, dimana **hari > 0** akan dirubah menjadi (tahun, bulan, hari) tahun **kabisat** diabaikan, dan mengembalikan tuple

B. Bilangan

3. Jika diberikan bilangan $n \geq 1$ sebagai input fungsi **generator(n)**, dimana fungsi ini akan menghasilkan list **lst**, yang berisi urutan angka yang dihitung dengan algoritma sbb:
 - a. Tambahkan **n** pada list **lst**
 - b. Lakukan loop selama **n** tidak sama dengan 1
 - 1) Jika $n = \text{genap}$ maka $x = n/2$, dan **x** ditambahkan pada list **lst**,
 - 2) Jika $n = \text{ganjil}$ maka $y = (3 * n) + 1$ dan **y** ditambahkan pada list **lst**.

Sehingga **generator(10) → [10, 5, 16, 8, 4, 2, 1]**

4. Berdasarkan algoritma pada soal no 3, buatlah fungsi **meanmax(n)**, yang mengembalikan tuple dengan 3 elemen, **(n, max, rerata)**, jika **n=10**, maka **max = 16**, **rerata = (10+5+16+8+4+2+1)/7 = 6.58**, sehingga **meanmax(10) → (10, 16, 6.571428571428571)**.

Petunjuk : boleh menggunakan fungsi **list.append()**

C. String

5. Buatlah fungsi untuk mencari karakter **c** dalam string **S**, dan mengitung jumlah kemunculannya. Fungsi bernama **Cari(c, S)**, dan akan menghasilkan tuple **(c, n)**, **c** karakter yang dicari, **n** jumlah **c** dalam **S**. Sehingga **Cari('o', "ohhalloo") → ('o', 3)**, Tidak boleh menggunakan fungsi **count()**.
6. Buatlah fungsi yang **mengitung jumlah kemunculan tiap karakter dalam string S** yaitu **chFrek(S)**, fungsi akan mengembalikan **dictionary** yang berisi pasangan **karakter (key)** dan **jumlah kemunculannya (value)**. Contoh **chFrek("Halloo") → {'H': 1, 'a': 1, 'l': 2, 'o': 2}**. Tidak boleh menggunakan fungsi **count()**.

Petunjuk : Gunakan **for..in** dan **if ..in**

D. Logika

7. Buatlah fungsi `Lucky(a,b,c)` yang menerima 3 nilai int, a b c, fungsi akan mengembalikan jumlahnya (a+b+c), Namun, jika salah satu dari nilai a, b atau c adalah **13** maka nilai di sebelah *kanannya* tidak dihitung jumlahnya. Jadi misalnya, jika b adalah 13, maka b dan c tidak dihitung, dan yang dikembalikan hanya a. Contoh `Lucky(1,3,3) → 7`, `Lucky(1,2,13) → 3`, `Lucky(9,13,13) → 9`, `Lucky(13,13,13) → 0`
8. Buatlah fungsi yang menerima 3 int, a b c, **dekatjauh(a,b,c)** dan mengembalikan **True** jika salah satu dari **b** atau **c** adalah "dekat" (selisih paling banyak 1), sedangkan yang lain "jauh", berbeda dari kedua nilai lainnya 2 atau lebih. Petunjuk: fungsi `abs(n)` menghitung nilai absolut suatu n. Contoh, `dekatjauh(1,2,10) → True`, `dekatjauh(1,2,3) → False`, `dekatjauh(4,5,3) → False`, `dekatjauh(-1,10,0) → True`.

E. List

9. Buatlah fungsi **ada22(list)** untuk memeriksa apakah terdapat angka 2 yang berurutan sejumlah 2 buah dalam list sebagai argument fungsi. Kembalikan **True** jika ada, dan **False** jika tidak. `ada22([4,2,4,2,2,5]) → True`, `ada22([2,1,2]) → False`, `ada22([]) → False`, `ada22([2,2]) → True`
10. Buatlah Fungsi `jumlah13(list)`, yaitu fungsi yang menjumlahkan elemen list, namun bila terdapat elemen bernilai 13, maka elemen tersebut dan 1 elemen sesudahnya tidak ikut dijumlahkan. Contoh `jumlah13([1, 2, 2, 1]) → 6`, `jumlah13([1, 2, 13, 2, 1, 13]) → 4`, `jumlah13([13, 1, 2, 13, 2, 1, 13]) → 3`.

Petunjuk : lakukan loop sepanjang list, dan jika `list[i]=13`, maka **update**, `list[i]` dan `list[i+1]` dengan 0, gunakan fungsi `sum()` untuk mencari jumlahan list sebagai hasil balik fungsi.

```
#file pap2drv.py
from pap2lib import *
def main(): # main program
    print(cmToM(96)) # (0, 96)
    print(cmToM(196)) # (1, 96)
    print(cmToM(1)) # (0, 1)
    print(cmToM(2196)) # (21, 96)
    print(hariToTahun(2535)) # (6, 11, 15)
    print(generator(10)) # [10, 5, 16, 8, 4, 2, 1]
    print(meanmax(10)) # (10, 16, 6.571428571428571)
    print(Cari('o', "ohalloo")) # ('o', 3)
    print(chFrek("halloo")) # {'H': 1, 'a': 1, 'l': 2, 'o': 2}
    print(lucky(3,4,5)) # 12
    print(lucky(3,13,5)) # 2
    print(dekatjauh(1,2,10)) # True
    print(dekatjauh(1,2,3)) # False
    print(ada22([4,2,4,2,2,5])) # True
    print(ada22([2,1,2])) # False
    print(ada22([])) # False
    print(ada22([2,2])) # True
    print(jumlah13([13, 1, 2, 13, 2, 1, 13])) # 3
    print(jumlah13([1, 2, 13, 2, 1, 13])) # 4
if __name__ == '__main__': # entry program
    main()

#File pap2lib.py
#Global definition
```

```
def cmToM(c):
    """
    Mengembalikan tuple (meter,centimeter) hasil konversi c dalam centimeter
    """
    Pass

def hariToTahun(hari):
    """
    Mengembalikan tuple (tahun, bulan, hari) hasil konversi hari dalam jumlah hari
    """
    pass

def generator(n):
    """
    Mengembalikan list hasil input n dalam algoritma :
    Selama n>1:
        Jika n genap maka n bagi 2
        Jika n ganjil maka n=(3*n)+1
    Boleh menggunakan fungsi append() dalam list
    """
    pass

def meanmax(n):
    """
    Mengembalikan tuple yang berisi nilai maksimum dan rerata yang dihasilkan oleh
    algoritma seperti fungsi generator() diatas:

    """
    pass

def chFrek(S):
    """
    Mengembalikan frekuensi karakter dalam string S

    """
    pass

def Cari(c,S):
    """
    Mengembalikan tuple yang berisi karakter c dan jumlahnya dalam string S
    """
    pass

def lucky(a, b, c):
    """
    Sesuai deskripsi dalam soal
    """
    pass

def dekatjauh(a, b, c):
    """
    Sesuai deskripsi dalam soal
    """
    pass

def ada22(nums):
    """
    Sesuai deskripsi dalam soal
    """
```

```
pass
```

```
def jumlah13(nums):  
    """  
    Sesuai deskripsi dalam soal  
    """  
    pass
```

```
#Output yang diinginkan  
(0, 96)  
(1, 96)  
(0, 1)  
(21, 96)  
(6, 11, 15)  
[10, 5, 16, 8, 4, 2, 1]  
(10, 16, 6.571428571428571)  
('o', 3)  
{'h': 1, 'a': 1, 'l': 2, 'o': 2}  
12  
3  
True  
False  
True  
False  
False  
True  
3  
4
```