

SD2201 – PEMROGRAMAN BERBASIS FUNGSI

LAPORAN AKHIR PRATIKUM



Nama : Lanang Almasyuri
NIM : 120450109
Kelas : RA
Prodi : Sains Data

**INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2022**

Exercise 1 sebelum UTS

Exercise 1 sebelum UTS

In [67]:

```

# Lanang Almasyuri Program exercise 1
# Author : Lanang Almasyuri
# NIM : 120450109
# Affiliation : Sains Data ITERA
# Program Description : Program to solve simple encryption password problem

def encrypt(pwd):
    splitpass = list(pwd)

    asciipass = list()
    for char in splitpass:
        asciichar = ord(char)
        asciipass.append(asciichar)

    encryptedpass = ""
    for num in asciipass:
        firstval = num//26 + 80
        secondval = num%26 + 80
        if firstval > secondval:
            thirdval = '-'
        else:
            thirdval = '+'

        encryptedpass = encryptedpass + chr(firstval) + chr(secondval) + thirdval

    return encryptedpass

def decrypt(pwd):
    splitpass = [pwd[i:i+3] for i in range(0, len(pwd), 3)]

    asciipass = list()
    for word in splitpass:
        firstval = ord(word[0]) - 80
        secval = ord(word[1]) - 80
        val = 26 * firstval + secval
        asciipass.append(val)

    password = ''
    for i in asciipass:
        char = chr(i)
        password = password + char

    return password

```

In [68]:

```

while True :
    jawab=input("apakah kamu ingin melakukan enkripsi(y/n)")
    if jawab == "y":
        text = input("masukkan text pasword anda`")
        print ("Text : " + text)
        print ("Cipher: " + encrypt(text))
    else:
        print("terimakasih")
        break

```

terimakasih

```
In [69]: text="anakanakcerdas2020"
print(encrypt(text))
```

Sc+TV+Sc+TS-Sc+TV+Sc+TS-Se+Sg+TZ+Sf+Sc+T[+Qh+Qf+Qh+Qf+

```
In [70]: text="Sc-TV-Sc-TS+T[-Sc-TQ+TV-T[-Sf-Sc-T\ -Sc-Qh-Qf-Qh-Qf-TS+Sg-Se-Sg-"
print(decrypt(text))
```

anaksainsdata2020kece

Exercise 2 sebelum UTS

```
In [71]: # Lanang Almasyuri Program exercise 2
# Author : Lanang Almasyuri
# NIM : 120450109
# Affiliation : Sains Data ITERA
# Program Description : Program to solve simple encryption password problem

# Python3 program to find sum of
# two large numbers.

# Function for finding sum of
# larger numbers

file1 = open("C:/Users/Lenovo/OneDrive/Desktop/KELAS/PBF/Tugas/tugas wajib/exercise2
line1 = file1.readlines()
file1.close

file2 = open("C:/Users/Lenovo/OneDrive/Desktop/KELAS/PBF/Tugas/tugas wajib/exercise2
line2 = file2.readlines()
file2.close

str1 = str(line1)
str2 = str(line2)

def findSum(str1, str2):

    # Before proceeding further,
    # make sure length of str2 is larger.
    if (len(str1) > len(str2)):
        t = str1;
        str1 = str2;
        str2 = t;

    # Take an empty string for
    # storing result
    str = "";

    # Calculate length of both string
    n1 = len(str1);
    n2 = len(str2);

    # Reverse both of strings
    str1 = str1[::-1];
    str2 = str2[::-1];

    carry = 0;
    for i in range(n1):

        # Do school mathematics, compute
        # sum of current digits and carry
        sum = ((ord(str1[i]) - 48) +
```

```

        ((ord(str2[i]) - 48) + carry));
    str += chr(sum % 10 + 48);

    # Calculate carry for next step
    carry = int(sum / 10);

    # Add remaining digits of Larger number
    for i in range(n1, n2):
        sum = ((ord(str2[i]) - 48) + carry);
        str += chr(sum % 10 + 48);
        carry = (int)(sum / 10);

    # Add remaining carry
    if (carry):
        str += chr(carry + 48);

    # reverse resultant string
    str = str[::-1];

    return str;

```

In [72]: `print (str1, "+", str2, "hasilnya adalah =", findSum(str1,str2))`

```

['9502561694858652150281747994108545943651521215096841995237040384498740803993469376
602031341619585763'] + ['21160686426961629349657890805309928053919005689789584962015
55855833896833372295507803936243187061092'] hasilnya adalah = 8531618630337554815085
24753707463953874904342178407580049143859624033263763736576488440596758480664685510

```

Exercise 3 sebelum UTS

In [73]:

```

def euler(t,h,y,dy,Func):
    d2y = Func(t,y,dy)
    y_next = y + (h * dy)
    dy_next = dy + (h * d2y)
    return ( y_next, dy_next )

def euler_cromer(t,h,y,dy,Func):
    d2y = Func(t, y, dy)
    dy_next = dy + (h * d2y)
    y_next = y + (h * dy_next)

    return (y_next, dy_next)

```

In []:

```

from solver import *

def cauchy_euler(params,Func):
    # Initial Condition
    t0 = params['t0']
    t_akhir = params['t_akhir']
    h = params['h']
    y0 = params['y0']
    dy0 = params['dy0']

    res_euler = []
    t = []
    step = int((t_akhir - t0) / h)

    for i in range(step):
        tm = (i + 1) * h
        (y_next, dy_next) = euler(tm, h, y0, dy0, Func)

```

```

        res_euler.append(y_next)
        t.append(tm)
        y0 = y_next
        dy0 = dy_next

    return (t,res_euler)

def cauchy_eulercromer(params,Func):
    # Initial Condition
    t0 = params['t0']
    t_akhir = params['t_akhir']
    h = params['h']
    y0 = params['y0']
    dy0 = params['dy0']

    res_euler_cromer = []
    t = []
    step = int((t_akhir - t0) / h)

    for i in range(step):
        tm = (i + 1) * h
        (y_next, dy_next) = euler_cromer(tm, h, y0, dy0, Func)
        res_euler_cromer.append(y_next)
        t.append(tm)
        y0 = y_next
        dy0 = dy_next
    return (t, res_euler_cromer)

```

from solver import from math import import matplotlib.pyplot as plt

g = 9.81 # gravitational Accelaration l = 1 # pendulum length k = 0 # velocity coeff u0 = 0.5 * pi
du0 = 0 t0 = 0 t_akhir = 4 h = 0.01 w0 = g/l

def Func(t,u,du): return -w0 sin(u) - kdu

res_euler = [] res_eulercromer = [] t = [] step = int((t_akhir - t0) / h)

for i in range(step): tm = (i + 1) * h (u_next, du_next) = euler(tm, h, u0, du0, Func)
res_euler.append(u_next) t.append(tm) u0 = u_next du0 = du_next

t = [] u0 = 0.5 * pi du0 = 0 d2u0 = Func(t0,u0,du0)

for i in range(step): tm = (i + 1) * h (u_next, du_next) = euler_cromer(tm, h, u0, du0, Func)
res_eulercromer.append(u_next) t.append(tm) u0 = u_next du0 = du_next

plt.title('Non Linear Pendulum h =0.01') plt.plot(t,res_euler,color='r', label = 'Euler')
plt.plot(t,res_eulercromer,color='g', label = 'Euler Cromer') plt.xlabel('t') plt.ylabel('u(t)')
plt.legend()

plt.show()

Exercise 4 Sebelum UTS

In [75]:

```

import numpy as np

skillsA_ = {'save': 81, 'tackle1': 79, 'passing': 78, 'tackle2':60, 'dribble1': 76,
skillsB_ = {'save': 86, 'tackle1': 80, 'passing': 81, 'tackle2':70, 'dribble1': 70,
mentalA_ = {'GK':80, 'DF':79, 'MD':78, 'ATK':77}
mentalB_ = {'GK':77, 'DF':78, 'MD':79, 'ATK':80}

```

```

kb = 100000
ka = 115000
b = 0
a = 0
t = 90
curr = "A"
def calc_skill(skill, mentallity):
    def alpha(mentallity):
        return np.random.uniform(0,0.025) * (mentallity/100)
    def beta():
        return np.random.uniform(0,0.025) * (ka/(ka+kb))
    return skill*(1-(alpha(mentallity)+beta()))

```

In [76]:

```

def md_(x,y):
    return calc_skill(x,y)

def atk_(x,y):
    return calc_skill(x,y)

def gk_(x,y):
    return calc_skill(x,y)

def df_(x,y):
    return calc_skill(x,y)

```

In [77]:

```

def matchStart():
    print('start...')
    print("=====SIMPLE FOOTBALL GAME=====")
    print('Pertandingan dimulai')
    fieldMid()

def resultCheck():
    global a, b
    return print(f'Skor pertandingan saat ini adalah {a} - {b}')

def stopBreak(x):
    x = 0
    return x

def resultFinal():
    global a,b
    return print(f'Hasil akhir pertandingan adalah {a} - {b}')

def fieldMid():
    global t, curr
    if t >= 0:
        if curr == "A":
            if md_(skillsA_.get('dribble1'), mentalA_.get('MD')) > md_(skillsB_.get(
                t -= 1
                print('Bola dioper ke penyerang tim A')
                print(f'waktu pertandingan tersisa {t} menit')
                curr = "A"
            else:
                t -= 2
                print('Bola diambil oleh penyerang tim B')
                print(f'waktu pertandingan tersisa {t} menit')
                curr = "B"
        else:
            if md_(skillsB_.get('dribble1'), mentalB_.get('MD')) > md_(skillsA_.get(
                t -= 1
                print('Bola dioper ke penyerang tim B')
                print(f'waktu pertandingan tersisa {t} menit')

```

```

        curr = "B"
    else:
        t -= 2
        print('Bola diambil oleh penyerang tim A')
        print(f'waktu pertandingan tersisa {t} menit')
        curr = "A"
    fieldAtk()
else:
    stopBreak(t)

def fieldAtk():
    global t, curr, a, b
    if t >= 0:
        if curr == "B":
            if atk_(skillsB_.get('dribble2'),mentalB_.get('ATK')) > df_(skillsA_.ge
            print('Penyerang tim B akan melakukan shooting')
            print(f'waktu pertandingan tersisa {t} menit')
            t -= 1
            if atk_(skillsB_.get('shoot'),mentalB_.get('ATK')) > gk_(skillsA_.ge
            t -= 2
            print('tendangan berhasil memasukkan gawang, gol bestie!')
            print(f'waktu pertandingan tersisa {t} menit')
            b += 1
            resultCheck()
            fieldMid()
            curr = "B"
        else:
            t -= 3
            print('sayang sekali, tendangan tidak berhasil memasuki gawang')
            print(f'waktu pertandingan tersisa {t} menit')
            curr = "A"
    else:
        t -= 2
        print('Bola diambil oleh pemain pertahanan tim A')
        print(f'waktu pertandingan tersisa {t} menit')
        curr = "A"
    else:
        if atk_(skillsA_.get('dribble2'),mentalA_.get('ATK')) > df_(skillsB_.get
        t -= 1
        print('Penyerang tim A akan melakukan shooting')
        print(f'waktu pertandingan tersisa {t} menit')
        if atk_(skillsA_.get('shoot'),mentalA_.get('ATK')) > gk_(skillsB_.ge
        t -= 2
        print('tendangan berhasil memasukkan gawang, gol bestie!')
        print(f'waktu pertandingan tersisa {t} menit')
        a += 1
        resultCheck()
        fieldMid()
        curr = "A"
    else:
        t -= 3
        print('sayang sekali, tendangan tidak berhasil memasuki gawang')
        print(f'waktu pertandingan tersisa {t} menit')
        curr = "B"
    else:
        t -= 2
        print('Bola diambil oleh pemain pertahanan tim B')
        print(f'waktu pertandingan tersisa {t} menit')
        curr = "B"
    fieldDf()
else:
    stopBreak(t)

def fieldDf():

```

```

global curr, t
if t >= 0:
    if curr == "B":
        if df_(skillsB_.get('passing'), mentalB_.get('DF')) > atk_(skillsA_.get(
            t -= 2
            print('Bola dioper ke pemain tengah B')
            fieldMid()
        else:
            t -= 1
            print('Bola diambil oleh penyerang tim A')
            fieldAtk()
    else:
        if df_(skillsA_.get('passing'), mentalA_.get('DF')) > atk_(skillsB_.get(
            t -= 2
            print('Bola dioper ke pemain tengah A')
            fieldMid()
        else:
            t -= 1
            print('Bola diambil oleh penyerang tim B')
            fieldAtk()
    else:
        stopBreak(t)

```

In [78]:

matchStart()

start....

=====SIMPLE FOOTBALL GAME=====

Pertandingan dimulai

Bola dioper ke penyerang tim A

waktu pertandingan tersisa 89 menit

Penyerang tim A akan melakukan shooting

waktu pertandingan tersisa 88 menit

tendangan berhasil memasuki gawang, gol bestie!

waktu pertandingan tersisa 86 menit

Skor pertandingan saat ini adalah 1 - 0

Bola dioper ke penyerang tim A

waktu pertandingan tersisa 85 menit

Bola diambil oleh pemain pertahanan tim B

waktu pertandingan tersisa 83 menit

Bola diambil oleh penyerang tim A

Penyerang tim B akan melakukan shooting

waktu pertandingan tersisa 82 menit

tendangan berhasil memasuki gawang, gol bestie!

waktu pertandingan tersisa 79 menit

Skor pertandingan saat ini adalah 1 - 1

Bola dioper ke penyerang tim B

waktu pertandingan tersisa 78 menit

Penyerang tim B akan melakukan shooting

waktu pertandingan tersisa 78 menit

tendangan berhasil memasuki gawang, gol bestie!

waktu pertandingan tersisa 75 menit

Skor pertandingan saat ini adalah 1 - 2

Bola dioper ke penyerang tim B

waktu pertandingan tersisa 74 menit

Penyerang tim B akan melakukan shooting

waktu pertandingan tersisa 74 menit

tendangan berhasil memasuki gawang, gol bestie!

waktu pertandingan tersisa 71 menit

Skor pertandingan saat ini adalah 1 - 3

Bola dioper ke penyerang tim B

waktu pertandingan tersisa 70 menit

Penyerang tim B akan melakukan shooting

waktu pertandingan tersisa 70 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 67 menit
Skor pertandingan saat ini adalah 1 - 4
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 66 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 66 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 63 menit
Skor pertandingan saat ini adalah 1 - 5
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 62 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 62 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 59 menit
Skor pertandingan saat ini adalah 1 - 6
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 58 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 58 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 55 menit
Skor pertandingan saat ini adalah 1 - 7
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 54 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 54 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 51 menit
Skor pertandingan saat ini adalah 1 - 8
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 50 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 50 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 47 menit
Skor pertandingan saat ini adalah 1 - 9
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 46 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 46 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 43 menit
Skor pertandingan saat ini adalah 1 - 10
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 42 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 42 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 39 menit
Skor pertandingan saat ini adalah 1 - 11
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 38 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 38 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 35 menit
Skor pertandingan saat ini adalah 1 - 12
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 34 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 34 menit

tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 31 menit
Skor pertandingan saat ini adalah 1 - 13
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 30 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 30 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 27 menit
Skor pertandingan saat ini adalah 1 - 14
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 26 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 26 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 23 menit
Skor pertandingan saat ini adalah 1 - 15
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 22 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 22 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 19 menit
Skor pertandingan saat ini adalah 1 - 16
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 18 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 18 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 15 menit
Skor pertandingan saat ini adalah 1 - 17
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 14 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 14 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 11 menit
Skor pertandingan saat ini adalah 1 - 18
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 10 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 10 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 7 menit
Skor pertandingan saat ini adalah 1 - 19
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 6 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 6 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa 3 menit
Skor pertandingan saat ini adalah 1 - 20
Bola dioper ke penyerang tim B
waktu pertandingan tersisa 2 menit
Penyerang tim B akan melakukan shooting
waktu pertandingan tersisa 2 menit
tendangan berhasil memasukkan gawang, gol bestie!
waktu pertandingan tersisa -1 menit
Skor pertandingan saat ini adalah 1 - 21

PERTEMUAN 7 : Higher Order Function and Map

In [22]: *# 1 list tuples of P*

```
P = 'akulupa'
print(*map(lambda p: (p[0]*2+1, p[1]), enumerate(P)))
```

```
(1, 'a') (3, 'k') (5, 'u') (7, 'l') (9, 'u') (11, 'p') (13, 'a')
```

In [23]:

```
# 2 petakan B menjadi List faktornya

B = 24

two = map(lambda b: b+1 if B % (b+1) == 0 else -1, range(B))

def dua(two):
    x = []
    for t in two:
        if t != -1:
            x.append(t)
    return x

print(dua(two))
```

```
[1, 2, 3, 4, 6, 8, 12, 24]
```

In [24]:

```
# nomor 3
A = [[3,4], [5,6]]
B = [[1,2], [7,8]]

C=list(map(lambda ra, rb:list(map(lambda raa,rbb: raa+rbb, ra, rb)), A, B))

def dett (C):
    return C[0][0]*C[1][1]-C[0][1]*C[1][0]

print(dett(C))
```

```
-16
```

PERTEMUAN 9 : Higher order Function: Filter

In [25]:

```
# Soal No 1
# Buat program untuk menghitung deret bilangan prima dari 2 hingga menggunakan HOF f

factor = lambda n : list(filter(lambda i: n%i!=0, range(1,n+1)))
primes = lambda n : filter(lambda i : len(factor(i)) == 2, range(1,n+1))
print(*primes(100))
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

In [26]:

```
# Soal No 2
# Latihan filter

employee = {
    'Nagao':35,
    'Ishii':30,
    'Kazutomo':20,
    'Saito':25,
    'Hidemi':29
}

print(employee.items())
```

```
dict_items([('Nagao', 35), ('Ishii', 30), ('Kazutomo', 20), ('Saito', 25), ('Hidem
i', 29)])
```

```
In [27]: filter_by_age = lambda age, employee: list( filter(lambda x:x [1] >= 25, employee.it
filtered_arr = filter_by_age
print(* map (lambda d:d[0], filter_by_age(25, employee)))
```

Nagao Ishii Saito Hidemi

PERTEMUAN 10 Higher Order Function - Reduce

```
In [28]: # nomer 1
# Buat fungsi mencari jumlah bilangan genap dari list L !

from functools import reduce as r

L = [2,1,9,10,3,90,15]

n_genap = lambda L: r(lambda a,b: a + 1 if b%2 == 0 else a, L, 0)
n_genap(L)
```

Out[28]: 3

```
In [29]: # nomer 2
# Buat fungsi untuk menghitung n! menggunakan reduce!
facto = lambda n: r(lambda a,b: a*b if b>1 else 1, range(1, n+1), 1)
for i in range(10+1):
    print( str(i) + '!=', facto(i) )
```

```
0!= 1
1!= 1
2!= 2
3!= 6
4!= 24
5!= 120
6!= 720
7!= 5040
8!= 40320
9!= 362880
10!= 3628800
```

```
In [30]: # nomer 3
# Hitung euclidian distance dari dua vektor berikut menggunakan higher order functio
X = [2,5,6,7,10]
Y = [-2,9,2,-1,10]

euclid = lambda X,Y: r(lambda a,c:a+c, map(lambda x,y: (x-y)**2, X, Y))**0.5
euclid(X,Y)
```

Out[30]: 10.583005244258363

```
In [31]: # nomer 4
# Terdapat dictionary employee berisi nama dan umur pegawai, lakukan reduce untuk me
employee = {
    'Nagao':35,
    'Ishii':30,
    'Kazutomo':20,
    'Saito':25,
    'Hidemi':29
}

cnt_emp = lambda lim,employee: r(lambda a,b: a+1 if b[1]>lim else a, employee.items(
cnt_emp(25,employee)
```

Out[31]: 3

In [32]:

```
# nomer 5
# Buatlah deret fibonacci menggunakan higher order function!
# Non Recursive menggunakan Reduce

fibonacci = lambda n: r(lambda a,b: a if b[0] <=1 else a + [ a[ b[0]-1 ] + a[ b[0]-2 ]], e
for i in range(10):
    print('Fibonacci',i,'->',fibonacci(i))

Fibonacci 0 -> [0]
Fibonacci 1 -> [0, 1]
Fibonacci 2 -> [0, 1, 1]
Fibonacci 3 -> [0, 1, 1, 2]
Fibonacci 4 -> [0, 1, 1, 2, 3]
Fibonacci 5 -> [0, 1, 1, 2, 3, 5]
Fibonacci 6 -> [0, 1, 1, 2, 3, 5, 8]
Fibonacci 7 -> [0, 1, 1, 2, 3, 5, 8, 13]
Fibonacci 8 -> [0, 1, 1, 2, 3, 5, 8, 13, 21]
Fibonacci 9 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

PERTEMUAN 11 - Recursion in FP

In [33]:

```
# Recursive
#Latihan
# Buat sebuah program untuk membuat deret fibonacci dari 0 hingga N dengan menggunak
fibonacci_rec = lambda n: 0 if n==0 else 1 if (n==1 or n==2) else fibonacci_rec(n-1) + fibonacci_rec(n-2)
deret_fibonacci_rec = lambda n: list( map(lambda i: fibonacci_rec(i), range(n+1)))
deret_fibonacci_rec(10)
```

Out[33]: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

Jadi kesimpulannya dalam membuat fibonacci, lebih baik menggunakan non recursive terlihat dari hasil running yang lebih cepat

PERTEMUAN 12 Purity and Immutability

In [34]:

```
#Latihan 1
#Ubah fungsiku menjadi pure function!
def fungsiku(L):
    def check_genap(l):
        return l % 2 == 0
    for i in range(len(L)):
        if check_genap(L[i]):
            L[i] = L[i]/2
        else:
            L[i] = L[i]*n+1
    return L
```

In [35]:

```
n = 3
L = [5,6,7,8]
print(fungsiku(L))
```

[16, 3.0, 22, 4.0]

In [36]:

```
print(L)
```

[16, 3.0, 22, 4.0]

In [37]:

```
n = 3
L = [5,6,7,8]
def fungsiku(L, n):
    return list(map(lambda x: x/2 if x%2==0 else x*n+1, L))
print(fungsiku(L,n))
print(L)
```

```
[16, 3.0, 22, 4.0]
[5, 6, 7, 8]
```

In [38]:

```
#Latihan 2
#Ubah fungsiku2 menjadi pure function!
def fungsiku2(L):
    def check_faktor(l):
        return l % n == 0
    for i in range(len(L)):
        if check_faktor(L[i]):
            L[i] = L[i]/2
        else:
            L[i] = L[i] * n + 1
    return L
```

In [39]:

```
n = 3
L = [5,6,7,8]
print(list(fungsiku2(L)))
print(L)
```

```
[16, 3.0, 22, 25]
[16, 3.0, 22, 25]
```

In [40]:

```
n = 3
L = [5,6,7,8]
def fungsiku2(L, n):
    return list(map(lambda x: x/2 if x%n==0 else x*n+1, L))
print(fungsiku2(L,n))
print(L)
```

```
[16, 3.0, 22, 25]
[5, 6, 7, 8]
```

In [41]:

```
#Latihan 3
#Apakah isi dalam tuple tup ada yang dapat diubah?
tup = ([3, 4, 5], 'myname')
tup[0]="lanang"
print("tup 2 :", tup)
```

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_17680\1313421453.py in <module>
      2 #Apakah isi dalam tuple tup ada yang dapat diubah?
      3 tup = ([3, 4, 5], 'myname')
----> 4 tup[0]="lanang"
      5 print("tup 2 :", tup)
```

TypeError: 'tuple' object does not support item assignment

Bisa dilihat diatas bahwa jika data bertipe tuple dia tidak bisa diubah atau immutable

PERTEMUAN 13 : Function Building Function

```
In [42]: #Latihan
from functools import reduce as r
keranjang = [
    {'Jumlah_Barang': 5, 'Harga': 10},
    {'Jumlah_Barang': 7, 'Harga': 20},
    {'Jumlah_Barang': 20, 'Harga': 4.5}
]
```

```
In [43]: def pajak_decorator(func):
    def inner(*args, **kwargs):
        res = func(*args, **kwargs)
        print('Sub Total: ', res)
        print('Pajak: ', res * 0.11)
        print('Total: ', res + res * 0.11)
        return res
    return inner

import time

def calc_time_decorator(fu):
    def inner(*args, **kwargs):
        waktu_awal = time.time()
        res = fu(*args, **kwargs)
        waktu_akhir = time.time()
        print('Waktu eksekusi: ', waktu_akhir - waktu_awal)
        return res
    return inner
```

```
In [44]: @calc_time_decorator
@pajak_decorator
def hitung_pembayaran_1(keranjang):
    return r(lambda a,b: a + (b['Jumlah_Barang'] * b['Harga']), keranjang, 0)

hitung_pembayaran_1(keranjang)
```

```
Sub Total: 280.0
Pajak: 30.8
Total: 310.8
Waktu eksekusi: 0.0
280.0
```

Out[44]:

```
In [45]: @calc_time_decorator
@pajak_decorator
def hitung_pembayaran_2(keranjang):
    s = 0
    for k in keranjang:
        s = s + k['Jumlah_Barang'] * k['Harga']
    return s * 1000
hitung_pembayaran_2(keranjang)
```

```
Sub Total: 280000.0
Pajak: 30800.0
Total: 310800.0
Waktu eksekusi: 0.0005629062652587891
280000.0
```

Out[45]:

Latihan Akhir

```
In [46]: import numpy as np
```

```
from random import randint
from functools import reduce as r
```

```
In [47]: def fungsim(a, b):
          M= np.random.uniform(a, b, size=(2, 2))
          return M
          M=fungsim(-1, 1)
          M
```

```
Out[47]: array([[ -0.27941827, -0.63652919],
                [ 0.77738588,  0.39804787]])
```

```
In [48]: def fungsiw(a, b):
          W= np.random.uniform(a, b, size=(3, 2))
          return W
          W=fungsiw(-1, 1)
          W
```

```
Out[48]: array([[ -0.87793445,  0.19590005],
                [-0.79849184,  0.58883958],
                [ 0.553691   ,  0.77717252]])
```

```
In [49]: def fungsih(a, b):
          H= np.random.uniform(a, b, size=(2, 1))
          return H
          H=fungsih(-1, 1)
          H
```

```
Out[49]: array([[ 0.1657142 ],
                [-0.26120124]])
```

```
In [50]: X=[1,0,9]
```

```
In [51]: import time

def calc_time_decorator(func):
    def inner(*args, **kwargs):
        start = time.time()
        res = func(*args, **kwargs)
        end = time.time()
        print('Time: ', end - start)
        return res
    return inner
```

```
In [52]: import math

def aktivasi(x):
    return (math.exp(x)-math.exp(-x))/(math.exp(x)+math.exp(-x))
def WTi(W,i):
    return list(map(lambda w: w[i], W))
def WT(W):
    return list(map(lambda i: WTi(W,i), range(0,len(W[0]))))
def XW(X,W):
    return map(lambda w:r(lambda x,y:x+y, map(lambda xx,ww:xx*ww, X,w)), WT(W))
def input_to_hidden(X,W):
    return map(lambda x: aktivasi(x), XW(X,W))

A=list(input_to_hidden(X,W))
```



```
def MTi(M,i):  
    return list(map(lambda m: m[i], M))  
def MT(M):  
    return list(map(lambda i: MTi(M,i), range(0,len(M[0]))))  
def AM(A,M):  
    return map(lambda m:r(lambda x,y:x+y, map(lambda aa,mm:aa*mm, A,m)), MT(M))  
def hidden_to_hidden(A,M):  
    return map(lambda a: a, AM(A,M))  
  
@calc_time_decorator  
def feed_forward(A,M,H):  
    return hidden_to_hidden(hidden_to_hidden(A,M),H)
```

```
In [65]: print(list(feed_forward(A,M,H)))
```

```
Time:  0.0  
[0.14474670464513756]
```

```
In [ ]:
```

PRAKTIKUM ! - Jurnal

```
In [1]: # kategori, harga per Lembar per hari
# input tanggal 2020-01-02
# input durasi 25
tanggal = input('Tanggal Pinjam: ')
durasi = int(input('Durasi Pinjam (hari): '))

kategoris = {
    1: 100,
    2: 200,
    3: 250,
    4: 300,
    5: 500,
}
```

Disini kita inputkan tanggal dengan = 2022-4-11 dan durasi dengan 24 hari.

```
In [2]: def dtl (s_tgl):
        return [ int(k) for k in s_tgl.split('-')]

def is_cm (tgl_p,d,c):
    return tgl_p[2]+ d > c

def thn_back (tgl_p,d,c):
    return tgl_p[0]+1 if ( is_cm(tgl_p,d,c) and tgl_p[1] == 12) else tgl_p[0]

def bln_back (tgl_p,d,c):
    return ( tgl_p[1] % 12 )+1 if is_cm(tgl_p,d,c) else (tgl_p[1])

def tgl_back (tgl_p,d,c):
    return tgl_p[2] + d - c if is_cm(tgl_p,d,c) else tgl_p[2] + d

def is_awal_abad(thn):
    return thn % 100 == 0

def kabisat (thn):
    return(is_awal_abad(thn) and thn % 400 == 0) or (not is_awal_abad(thn) and thn %

def dec_c(t):
    return 30 +( t[1]%2 if t[1]<= 8 else abs((t[1]%2)-1)) if t[1] != 2 else(29 if k

def wkt_kembali (tgl_p,d):
    return [thn_back(tgl_p,d, dec_c(tgl_p)),bln_back(tgl_p,d, dec_c(tgl_p)),tgl_back
```

```
In [12]: tgl_p = dtl(tanggal)
wkt_kembali(tgl_p,durasi)
```

```
Out[12]: [2022, 5, 5]
```

```
In [13]: sewaan_all = [ [1,5], [2,3], [3,0], [4,1], [5,2] ]

def calc_biaya_per_kategori(kategoris, sewaan):
    return sewaan[1] * kategoris.get(sewaan[0])

def calc_all_biaya(kategoris, sewaan_all, durasi):
    return sum([calc_biaya_per_kategori(kategoris, sewaan) for sewaan in sewaan_all])
```

```
In [14]: calc_all_biaya(kategoris, sewaan_all, durasi)
```

```
Out[14]: 57600
```

PRAKTIKUM 2 - Jurnal

In [84]:

```
#1

n = 5

def ulangi_120450109(a):
    listan=list(map(lambda x: a , range(1, a+1)))
    return listan

ulangi_120450109(5)
```

Out[84]: [5, 5, 5, 5, 5]

ulangi_120450109 adalah sebuah fungsi yang dibuat memiliki input sebuah bilangan skalar a, dan mengeluarkan output berupa vektor 1xn dengan seluruh elemen nya adalah a

In [25]:

```
#2

n = 5
iterable= list(range(1,n+1))

def deret(n):
    return ((-1)**(n+1)) * (1/(2**n))

print(list(map(deret, iterable)))
```

[0.5, -0.25, 0.125, -0.0625, 0.03125]

code di atas adalah membuat fungsi deret yang berisi fungsi perhitungan untuk membuat sebuah deret bilangan sesuai soal dengan inputnya n sebagai panjang deretnya

In [108...]

```
#3

from functools import reduce as r

L=list(map(deret, iterable))

def add(a,b):
    res=a+b
    print( "a=",a, "+ b=",b, "res=",res)
    return res

print(r(lambda a, b: a+b, L))
print(r(add,L))
```

0.34375

a= 0.5 + b= -0.25 res= 0.25

a= 0.25 + b= 0.125 res= 0.375

a= 0.375 + b= -0.0625 res= 0.3125

a= 0.3125 + b= 0.03125 res= 0.34375

0.34375

code diatas adalah sebuah cara untuk menghitung jumlah deret dari bilangan yang telah kita buat disoal sebelumnya. kita bisa melakukannya dengan langsung cetak seperti fungsi print yang pertama, atau kita bisa menggunakan fungsi add untuk dapat menunjukan nilai dari masing-masing a dan b terlebih dahulu baru kemudian menampilkan result nya per iterasi. kita dapat melakukannya dengan menggunakan reduce saja reduce wajib di definisikan menggunakan 2 argumen yaitu 1 akumulator dan 1 lagi elemen nya

```
In [155... file1 = open("C:/Users/Lenovo/OneDrive/Desktop/KELAS/PBF/praktikum/2/DNA.txt", "r")
dat = file1.read()
dat= dat[:-1] ##memotong index /n dipaling belakang data dat
seq="ACT"
dat
```

```
Out[155... 'TGTCTTCCGGCTGAGCGGTTCTAACCAGCAGACTGATACTGGTCGAATATCGACGGGCAAGAGCCCTGGGATTGATGCGTTT
CACCATGCGCGTCTCAGTGACGGCAGGAATGCAGAGCTTACTTCAAAGTAGTTACTGGCAAAAAATACAAATTTTTTCGATCGA
CCTTGAGTTTATTTCATTACCGCACAGTCTTTTACCGCACCTGTTACCGCACATCCGTAAGTTTACCGCACGTTACCGCACTACC
TCTCTATATTACCGCACTTCGTTTACCGCACGCTGAGGAACGGTTACCGCACTTACCGCACCACAAGGTGCGTGCTCTGTTATT
ACCGCACCACCATTACCGCACGCACTTTTATTACCGCACCAGGGCAGGCCACGTAGGGTAGCGTCTGTTCTACTGTATTGCGG
CGACGGTCTGAATTTACCGCATTACCGCACCACCTCGTTAGCTTACCGCACCTAGGGTTGTTACCGCACGACTTACCGCACAGCC
GTTACCGCACGTGTTACTTGACGCTCTAACTCCCACCTCATATCAGTCTTATTACCGCACACTGGGCTTACCGCACCCGCACCTT
AAGTAGGCAGTTACCGCACGTATTACCGCACGTAATTACCGCACACCTGTAAAGGCAGGGTAAAGTACAGACTTACCGCTTACC
GCACGGTTGCACCACGACAAATCTAACGTTAGGTACGTTACCGCACGGGAAATTACCGCACTCCAGGGTTTTACCGCACAGATA
TCCATTTCGGGAATGTGACCCCTGGAGTGAGATTGTGCGAAAGATACGGAGTTTTCAAGGGCACACCCAGCTATGTTATTAAGCG
TTACAGTGCGCGCTGCATCATGTCAATGTTGAGTTCATTCTCTATCTTGCTATGTACGAACCCTCGTTAAGAGGGAGTAAGCGA
TCTTTTGACAAAATCGTATGCATGTAGGCGAGGCAATGCCGATTACATTGAACGGCGGGACTTTTCGTATGAGACACCGCGGTT
GAAATATTTTTTATGCAAGAGCGGGATTGGGCGGAAGGAGACTTAACGCAGTGCTAGCACTGTTAACTGCGGCATGGCCGGA
TGGACTACCTATTTTGCAGCTCCAGCGTTTGAGTTCACGTACTGACGGAACAGTCCCGAGATAGGCCATGTGGTTCGATCCAG
TGAGAAATGAGACTCGAGATGCCGGTACCGGTAGCATCACCACATTGCTCCAGTATGATATCAGTCTTCACTGTACGAATTA
TGCAGCGATCTTGAAGAGAGTTATTCATCTCTTATCACCTGACAATAAATCAATTTACCAGTCAAATCTCTTTAACATCGTGC
CGAACTGCGATGCGTCGTAGTCTAGATTAGGATATATTTTCTAGCTGGCTTCGATGATTGGCTGTACGCTAAGGTGATTGAAT
TTCGATCTGCATTGGAGCTGTACCCACCTTGCATGGCATTGACAGCCTAAAGCGTGAAGAATGCAATACAGTGCAGAAAAA
TAACGGGCTCGATAACGTTCCAAGATTCTGACTTAACGACGGCTAGCGAGCGAGTCATAATCCCGTCCACACCGGGCAATCGG
GTCGGAGTGGAAGGGCGGGATTTTATTATTACGTGACGCAGATCTCCGTGCTACTATACTCACATCCTCTCTGTAGATAAAGT
TATACCAACCTCCATATTCTTCTTACGCTAAGTTCGGGCTATCCGAGTCTCGGCCATAGCAGGAGCACTTTAAGGGAAGTCTT
ATTGCCGAATACAGTACGTTCCCCGAATATGTTATACTACCCAAATATGTTAATATGTTATATGTTAAAACGCAGTGTGGGA
ATATGTTAATATGTTATGTGAATATGTTAAAATATGTTAAAATATGTTAACGATGTTAGCCGTGATAAATATGTTATTAACGGC
GTGCGTTAATATGTTAGCGACGACTGGGGGTCAATATGTTAGCCAACTTCTCAATATGTTAACGGTTAATATGTTAGTTAAG
ATCAATAAATATGTTAGCTACGTAGACAATAAAGCATAAGCAATATGTTATAATATGTTAGACAGTCTCTAACCGATAATAT
GTTAAGGCATACTTAACCAGCGAATGACAGAAATATGTTAATATGTTAAATAAATATGTTAGATAAATATGTTACGATATTACCC
GCACATTGCTCCGAATATGAATATGTTAAGGTGGTTCTCCGTATTTAATATTGTGAGAGATAGCTTGTGAGAGATGTTGTTGTG
AGAGAGCTGTGAGAGCTTTGCGAGCCTTTAAATATTGTGAGAGTTATGTAGTCGGCTGTGAGAGATGTGTGAGAGAGTTAAAT
ATGTTAGTTAGCTGAGAGCTACGCTGTGAGAGGCGAATTGACGTAGTGCTTTTTGTTCTGTGAGAGATGTGTGAGAGTGTGAGA
GCGCGTGTGAGTGTGAGAGTATTGTGCATGGTCCAGTAAGATGTGAGAGTGTGAGAGTGATCTAACGCTATGTGTGAGAGAGG
GTGTGAGAGGCTGCGTATGAAGCACAAATGTGAGAGTGTGAGAGATACGTTAAGAGCCCGGAAGCTCGGCATCATAAGCTGAG
CAGATTCAATGTGAGAGGGCGAGCCGACGGTAGGCTGTGAGAGTCATTATTGTGAGAGTCGCGTGGTGTGAGAGTCCATTTTA
TGTGAGATGTGAGAGCTCTGGGGCTGTGATGTGAGAGAGTACGCCGAAGCGTGTGAGAGTCTGTGAGAGATTCGGAGGTCTGG
ATGACATTGTGAGAGCCTGCTTACGCGACGTGATGAACGCGACCGACTAGCGACCGCCACTACTACTCGCAGTTGGTCTAGAG
GCATTGCTTTACTGAAATACGCAGGATGCTTATGACGCTCGCGCAATACATCGCGCTCGCACTGTATGTCGTTTCACTTAAT
CCTAAAGCTCAAATATAACGAAAAAGAGAAATTAGGACGACCGAGGGTCTGCTCCGGTGGTTTTACGACTTCGCCAATGGC
GTGCTGCGTCGAAATGTGCTCAAAGCCCCGTAAAGCTCAGACACCATGCAGGAATGGGAATGTGTACCCAGAGATCCCTAGTAA
GAGAGATCCAAGACTTAAAGCCGTTCCGAGAGAGATCTAATCACTAGAGATCTTAACACCAATAGAGATCCTCTAAGAGATCAG
TAGAGATCGCTTTTCAGAGATAGAGATCACTCACCAGAGAGATCTTACAGTTTGATATGTCAGTTCGGTTAAAAGCAGAGATCGT
CTGCAGAGATCGGTAGCGTAGAGATCCCGTGTGCTACAAAACTTAGAGATCAGATCGCGCCTCGAACTGTACTTAGAGATCTA
CATTATCTAAGAGAGAGATCAGAGATCACAAGGCCACACACGACAAAGTTAGAGATCTACACACGATAGGTGGTGCCGAACCTG
AGAGATCCGGGTTTTGAGAGAGATCAAGAGATAGAGATCGTTAGAGAAGAGATCTAGAGATCGCACGGGTTTTGGAGAGATCGT
TCGGGTTTTGTCGGGAAGAGATTAATGCGGTGAGTTAATGCGGGTATAATGCGGCAGATAATGTAATGCGGTCTAATGTAATGC
GGGAGATAATGCGGTGATGAACTTAATGCGGCTAATGCGGTATAATGCGGTGCAACGCTAATGCGGAGCTAATGCGGGCGTAAC
ATAATGTAATGCGGTTGTCAATATTGTTTTCAATATTAATATTCAATTACAATATTCAAACGCAATATTAACGCGCCGTAATG
CGGGTCAATCAATATTTTCGTAATGCGGGGTTAATGCGGTTTTCAATATTATCGGTAATGCGGGAGCTGGCAATATTGTTTTT
GGTAATGCGGTTTTAATAATGCGGGGCGACAATATTGGGTAATGCGGATATTTATCAATATTGTGTTCAATATTTAACACAATA
TTTGCCGTAGGTATGACCTAATTAATGCGGATATTAGGGGCCAATTAATGCGGATCGTAATGCGGGTGGGGCTTATAACAATTA
ATGCGGTCAATATTACTAATGCTAATGCGGCGGACTACAATATTTACAAAAGACTACCAATAATGCGGATAATGCGGTCAATAA
TGCGGAAGATAACGCGGCAATATTGCCCCGACAATATTTGACTACACAAGACTACACAATATTCGGTTATTCTGTGCCAACGCCA
GGTCAATGCGTCGAACCAATATTCTTGATTGTGATGCAGACTACACGACTACAATATTTACCCCGGGACTACATATCCACGAC
TACAGGGCGGAGACTACATAGGGACTACAGACTACAACAATTATGGTCACATTAACCTCTGCCCGGCGGCTCTTCCCTAAATCTCA
CGTGATGGACTAGACTACACCGACTACAACATACTTTGCAACGACTACAGTACGTTAAGACTACAGGATTACAGACTACACTTGA
TTTCCTGACTACACTTCTGACAACCCGCACATTGCCCGCTAACTCTGATGGCCCCAGAGACTACATACCATCGAGCGCGACTA
CAGGACTACAGCCGTAGACCTTTTAGACTACACGCCAGGGCCAACTGACACGGATAAGGTCTTTGCCCGCAAGTGCTCGCCGA
ATGTGATTAATCTCAACATTCGACCTGCAAGAGCACACGCATTGATATGGGTATAAGGAAGATCTCGTCCAGCTATAATGTA
CAACATTTCCCGTCATGACTTGCTACATAAACAGAATAAGACGTGACGTGCGCAATATAAGACGTACTCGATTGACCGTAAAA
TTTTCTAAGAACCAAAATAAGACGTAAGACGTTCACTTAATAAAGACGTAGGGCGTTACCGATAAGACGTTAAGACGTGGATC
```

```
GCCATCGCCCGTGAGTCGCTCTCCCGCATAAGACGTTAAGACGTCCCAATAGTGCTCCCTACACTTTACCGGTGGTAGATAAGA
CGTAGACGTTATAAGACGTCGGGTAAATATAAGACGTTATTTCCCAATAAATAAGACGTAATCCCTGTAACACTGGAAGTGATAA
GACGTTTGTCTAACATAAGACGTTGTAAGTGCCTAACCTGATAAGACGTTTTAAAAAGTACTATAAGACGTTTCGAGGAATG
AGACCATAAGACGTCGTCCTCCCTCAGCACTGAATTTTTTCGAAGATAAGACGTAAGACGTTTGGTTTATCGTTAGAAATAA
GACGTACGTTTATAAGACGTAATGGTCATAAGACGTACGTTAAGACGTAATAAGACGTTATCCATCCCAAAATTACACGTCAGA
AATCATGGCAACCGCGTGATGGAAGAGAGTAGCAACCGACTACATACAGTATACTGTGGGCAGACTCGTTTGTACACCAACAC
TTCCGCCGCCATTATTAATACGATTGGTGCTTTACGCATCTTGATGACCATGGTTACTCACCTCGGGTGCTGACCCGCCTGTC
TCCTATGACGTCGGGCTCCACTACGGCCCCGTTTCGACAGATAGGGGGGAGTTGACCTCGAATGCGGGTTACTTCGCCTGCCTT
TCGACGAATCGGTATGGCTAGCTTGGACAAGTATAGGATTGGTCTTTCAAGCTGCACTGTTTTGCAGCTTCTAGCGAGATAAGG
CTGAAGCCTCCAGCGATATTGTCCAGTTGGAAGAAAGTTGGAAGAAATGGGGGTTTGGAAAAAGAAAACGCCCGGGTTACACCG
GGGACATAATTGGAAAAAACAGTTGGAAAAAGCTTAGAAAGCTTGGAAAAAGTCTTGGGAACAATTATTGGAAAAACGATGGG
CGACTGAGAGTTGGAAAAAAAATTGGAAAAATTTGGAAAAACGTGGCTTTGGAAAAATGGAAAAAGATTGGAAAAATTGGAAAA
ACTTTTGAATTGGAAAAAAAAGCCACTGCGGGTGCTTTGGAAAAAATTTGGAAAAAAGTAGCAAAGCGGCATTCTGAGAGA
TTGGAAAAACGTGCTAAGCTTCTTTGGAAAAAGAATTGGAAAAAAGCGCACCCTCAGGAAGACATGTCTGGCACTTTAG
CGTTAAAGTTTGGAAAAAAGCTCTCCACATTTGGAAAAATGGAAAAAGAATCGGTTAGAGCGGCACGTGTCATATTGGAAAAA
TACTCAGCGCTTAGCAGTTGGAAAAAATGATGACTATGTTTGGAAAGACAAGGAGAAAGTCTCCGAACAACATCCATGACAAG
GAGGAGGCTGGACAAGGATTGAGGCTGTTTGGACAAGGAGGAGGACGACAAGGAAGGACTGTTTGGAGCTGGACAAGACAAGGACTG
TTGACAAGGACAGGACAAGGACGAAAGGCTGTTTGGAGGACAAGGAAGGACAGGCTGTTTGGAGGACGAGGACGACAAGGAAGGC
TGTTTGGAGCTGTTTGGAGGACGAGGAAGGATGTTTGGACAAGGAGGACGAGGCTAGGACGACGAAGAGGACGACTGTTTGGAGC
TGTAGGACGAGGACGAAAGGATGTTTGGACAAGGAGGAGAGGAGGACGAGGAAGGACGAAAGGAGACAAGGAGACAAGGAGACAAG
GAAGGACGAAGGACGAAGGACGAAGGACGAAGGACGAAGGACGAAGGACGAAGGACGAGGACGAGGACGAGGACGAAGGACGAAGGA
CAGGACGAAGGACGAAGGACGAAGGACGACAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCA
TCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCA
TCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCAATCATCA
```

In [111... type(dat)

Out[111... str

code diatas adalah code untuk berfungsi bagaimana membaca file txt yang ada didalam folder dengan memasukkan hasilnya berupa string ke dalam variabel dat

In [116...

```
def append_n(dat, i, n):
    return r(lambda x, y: x + y, [dat[i:i+n]])
append_n(dat, 0, 3)
```

Out[116... 'TGT'

bisa dilihat diatas, kita dapat membuat suatu pola dengan fungsi diatas sesuai panjang n dengan indeks awal dari i=0

In [136...

```
def remap(dat, seq):
    return map(lambda x: append_n(dat, x, len(seq)), range(0, len(dat)))
list(remap(dat, "ACT"))
```

Out[136...

```
['TGT',
'GTC',
'TCT',
'CTT',
'TTC',
'TCC',
'CCG',
'CGG',
'GGC',
'GCT',
'CTG',
'TGA',
'GAG',
'AGC',
```

'GCG',
'CGG',
'GGT',
'GTT',
'TTC',
'TCC',
'CCT',
'CTA',
'TAA',
'AAC',
'ACC',
'CCA',
'CAG',
'AGC',
'GCA',
'CAG',
'AGA',
'GAC',
'ACT',
'CTG',
'TGA',
'GAT',
'ATA',
'TAC',
'ACT',
'CTG',
'TGG',
'GGT',
'GTC',
'TCG',
'CGA',
'GAA',
'AAT',
'ATA',
'TAT',
'ATC',
'TCG',
'CGA',
'GAC',
'ACG',
'CGG',
'GGG',
'GGC',
'GCA',
'CAA',
'AAG',
'AGA',
'GAG',
'AGC',
'GCC',
'CCC',
'CCT',
'CTG',
'TGG',
'GGG',
'GGA',
'GAT',
'ATT',
'TTG',
'TGA',
'GAT',
'ATG',
'TGC',
'GCG',

'CGT',
'GTT',
'TTT',
'TTC',
'TCA',
'CAC',
'ACC',
'CCA',
'CAT',
'ATG',
'TGC',
'GCG',
'CGC',
'GCG',
'CGT',
'GTC',
'TCT',
'CTC',
'TCA',
'CAG',
'AGT',
'GTG',
'TGC',
'GCA',
'CAG',
'AGG',
'GGC',
'GCA',
'CAG',
'AGG',
'GGA',
'GAA',
'AAT',
'ATG',
'TGC',
'GCA',
'CAG',
'AGA',
'GAG',
'AGC',
'GCT',
'CTT',
'TTA',
'TAC',
'ACT',
'CTT',
'TTC',
'TCA',
'CAA',
'AAA',
'AAC',
'ACT',
'CTA',
'TAG',
'AGT',
'GTT',
'TTA',
'TAC',
'ACT',
'CTG',
'TGG',
'GGC',
'GCA',
'CAA',

'AAA',
'AAA',
'AAA',
'AAA',
'AAT',
'ATA',
'TAC',
'ACA',
'CAA',
'AAA',
'AAT',
'ATT',
'TTT',
'TTT',
'TTT',
'TTT',
'TTC',
'TCG',
'CGA',
'GAT',
'ATC',
'TCG',
'CGA',
'GAC',
'ACC',
'CCT',
'CTT',
'TTG',
'TGA',
'GAG',
'AGT',
'GTT',
'TTT',
'TTA',
'TAT',
'ATT',
'TTC',
'TCA',
'CAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACA',
'CAG',
'AGT',
'GTC',
'TCT',
'CTT',
'TTT',
'TTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACC',
'CCT',

'CTG',
'TGT',
'GTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACA',
'CAT',
'ATC',
'TCC',
'CCG',
'CGT',
'GTA',
'TAA',
'AAG',
'AGT',
'GTT',
'TTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGT',
'GTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACT',
'CTA',
'TAC',
'ACC',
'CCT',
'CTC',
'TCT',
'CTC',
'TCT',
'CTA',
'TAT',
'ATA',
'TAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACT',
'CTT',
'TTC',
'TCG',

'CGT',
'GTT',
'TTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGC',
'GCT',
'CTG',
'TGA',
'GAG',
'AGG',
'GGA',
'GAA',
'AAC',
'ACG',
'CGG',
'GGT',
'GTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACT',
'CTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACC',
'CCA',
'CAC',
'ACA',
'CAA',
'AAG',
'AGG',
'GGT',
'GTG',
'TGC',
'GCG',
'CGT',
'GTG',
'TGC',
'GCT',
'CTC',
'TCT',
'CTG',
'TGT',
'GTT',
'TTA',
'TAT',
'ATT',
'TTA',

'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACC',
'CCA',
'CAC',
'ACC',
'CCA',
'CAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGC',
'GCA',
'CAC',
'ACT',
'CTT',
'TTT',
'TTT',
'TTA',
'TAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACC',
'CCA',
'CAG',
'AGG',
'GGG',
'GGC',
'GCA',
'CAC',
'ACA',
'CAG',
'AGC',
'GCC',
'CCA',
'CAC',
'ACG',
'CGT',
'GTA',
'TAG',
'AGG',
'GGG',
'GGT',
'GTA',
'TAG',
'AGC',
'GCG',
'CGT',

'GTC',
'TCG',
'CGT',
'GTT',
'TTC',
'TCT',
'CTC',
'TCA',
'CAC',
'ACT',
'CTG',
'TGT',
'GTA',
'TAT',
'ATT',
'TTG',
'TGC',
'GCG',
'CGG',
'GGC',
'GCG',
'CGA',
'GAC',
'ACG',
'CGG',
'GGT',
'GTC',
'TCG',
'CGT',
'GTA',
'TAA',
'AAT',
'ATT',
'TTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACC',
'CCA',
'CAC',
'ACT',
'CTC',
'TCG',
'CGT',
'GTT',
'TTA',
'TAG',
'AGC',
'GCT',
'CTT',
'TTA',
'TAC',

'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACC',
'CCT',
'CTA',
'TAG',
'AGG',
'GGG',
'GGT',
'GTT',
'TTG',
'TGT',
'GTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGA',
'GAC',
'ACT',
'CTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACA',
'CAG',
'AGC',
'GCC',
'CCG',
'CGT',
'GTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGT',
'GTG',
'TGT',
'GTT',
'TTA',
'TAC',
'ACT',
'CTT',
'TTG',
'TGA',
'GAC',
'ACG',
'CGC',
'GCT',

'CTC',
'TCT',
'CTA',
'TAA',
'AAC',
'ACT',
'CTC',
'TCC',
'CCC',
'CCA',
'CAC',
'ACT',
'CTC',
'TCA',
'CAT',
'ATA',
'TAT',
'ATC',
'TCA',
'CAG',
'AGT',
'GTC',
'TCT',
'CTT',
'TTA',
'TAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACA',
'CAC',
'ACT',
'CTG',
'TGG',
'GGG',
'GGC',
'GCT',
'CTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACC',
'CCC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACC',
'CCT',
'CTT',
'TTA',
'TAA',
'AAG',
'AGT',
'GTA',

'TAG',
'AGG',
'GGC',
'GCA',
'CAG',
'AGT',
'GTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGT',
'GTA',
'TAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGT',
'GTA',
'TAA',
'AAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACA',
'CAC',
'ACC',
'CCT',
'CTG',
'TGT',
'GTA',
'TAA',
'AAA',
'AAG',
'AGG',
'GGC',
'GCA',
'CAG',
'AGG',
'GGG',
'GGT',
'GTA',
'TAA',
'AAA',
'AAG',
'AGT',
'GTA',
'TAC',
'ACA',

'CAG',
'AGA',
'GAC',
'ACT',
'CTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCT',
'CTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGG',
'GGT',
'GTT',
'TTG',
'TGC',
'GCA',
'CAC',
'ACC',
'CCA',
'CAC',
'ACG',
'CGA',
'GAC',
'ACA',
'CAA',
'AAA',
'AAT',
'ATC',
'TCT',
'CTA',
'TAA',
'AAC',
'ACG',
'CGT',
'GTT',
'TTA',
'TAG',
'AGG',
'GGT',
'GTA',
'TAC',
'ACG',
'CGT',
'GTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACG',
'CGG',
'GGG',

'GGA',
'GAA',
'AAA',
'AAT',
'ATT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACT',
'CTC',
'TCC',
'CCA',
'CAG',
'AGG',
'GGG',
'GGT',
'GTT',
'TTT',
'TTT',
'TTA',
'TAC',
'ACC',
'CCG',
'CGC',
'GCA',
'CAC',
'ACA',
'CAG',
'AGA',
'GAT',
'ATA',
'TAT',
'ATC',
'TCC',
'CCA',
'CAT',
'ATT',
'TTC',
'TCG',
'CGG',
'GGG',
'GGA',
'GAA',
'AAT',
'ATG',
'TGT',
'GTG',
'TGA',
'GAC',
'ACC',
'CCC',
'CCC',
'CCT',
'CTG',
'TGG',
'GGA',
'GAG',
'AGT',
'GTG',
'TGG',

'GGA',
'GAG',
'AGT',
'GTT',
'TTG',
'TGT',
'GTG',
'TGC',
'GCG',
'CGA',
'GAA',
'AAA',
'AAG',
'AGA',
'GAT',
'ATA',
'TAC',
'ACG',
'CGG',
'GGA',
'GAG',
'AGT',
'GTT',
'TTT',
'TTT',
'TTC',
'TCA',
'CAA',
'AAG',
'AGG',
'GGG',
'GGC',
'GCA',
'CAC',
'ACA',
'CAC',
'ACC',
'CCC',
'CCA',
'CAG',
'AGC',
'GCT',
'CTA',
'TAT',
'ATG',
'TGT',
'GTT',
'TTA',
'TAT',
'ATT',
'TTA',
'TAA',
'AAG',
'AGC',
'GCG',
'CGT',
'GTT',
'TTA',
'TAC',
'ACA',
'CAG',
'AGT',
'GTG',
'TGG',

'GGC',
'GCC',
'CCG',
'CGC',
'GCT',
'CTG',
'TGC',
'GCA',
'CAT',
'ATC',
'TCA',
'CAT',
'ATG',
'TGT',
'GTC',
'TCA',
'CAA',
'AAT',
'ATG',
'TGT',
'GTT',
'TTC',
'TCA',
'CAG',
'AGG',
'GGT',
'GTC',
'TCA',
'CAT',
'ATT',
'TTC',
'TCT',
'CTC',
'TCT',
'CTA',
'TAT',
'ATC',
'TCT',
'CTT',
'TTG',
'TGC',
'GCT',
'CTA',
'TAT',
'ATG',
'TGT',
'GTA',
'TAC',
'ACG',
'CGA',
'GAA',
'AAC',
'ACC',
'CCC',
'CCT',
'CTC',
'TCG',
'CGT',
'GTT',
'TTA',
'TAA',
'AAG',
'AGA',
'GAG',

'AGG',
'GGG',
'GGA',
'GAG',
'AGT',
'GTA',
'TAA',
'AAG',
'AGC',
'GCG',
'CGA',
'GAT',
'ATC',
'TCT',
'CTT',
'TTT',
'TTT',
'TTG',
'TGA',
'GAC',
'ACA',
'CAA',
'AAA',
'AAA',
'AAT',
'ATC',
'TCG',
'CGT',
'GTA',
'TAT',
'ATG',
'TGC',
'GCA',
'CAT',
'ATG',
'TGT',
'GTA',
'TAG',
'AGG',
'GGC',
'GCG',
'CGA',
'GAG',
'AGG',
'GGC',
'GCA',
'CAA',
'AAT',
'ATG',
'TGC',
'GCC',
'CCG',
'CGA',
'GAT',
'ATT',
'TTA',
'TAC',
'ACA',
'CAT',
'ATT',
'TTG',
'TGA',
'GAA',
'AAC',

```
'ACG',
'CGG',
'GGC',
'GCG',
'CGG',
'GGG',
'GGA',
'GAC',
'ACT',
'CTT',
'TTT',
'TTT',
'TTC',
'TCG',
'CGT',
'GTA',
'TAT',
'ATG',
'TGA',
'GAG',
'AGA',
'GAC',
'ACA',
'CAC',
'ACC',
'CCG',
...]
```

fungsi remap diatas digunakan untuk memisahkan atau split semua huruf didalam variabel dat sesuai panjang pola dari variabel seq yang kita inisialisasi. seperti pada code diatas kita isi seq dengan "ACT" berarti panjang pola yang tercipta adalah 3 huruf.

In [150...

```
def count_mer(dat, seq):
    return r(lambda x,y :x + (1 if y==seq else 0), remap(dat,seq), 0)
count_mer(dat, "ACT")
```

Out[150...

106

fungsi count_mer diatas berfungsi untuk mengitung jumlah pola yang sesuai dengan seq. Metode yang berjalan adalah dengan setiap iterasi jika menepukan pola sesuai seq pada remap yang telah kita buat sebelumnya maka nilai dari x akan ditambah satu.

In [144...

```
sequences=["A", "AT", "GGT", "AAGC", "AGCTA"]

def count_all(dat, sequences):
    return map(lambda x: count_mer(dat,x), sequences)

res=count_all(dat,sequences)
print(*res)
```

2112 557 77 22 5

sedangkan dengan fungsi count all ini kita dapat menhhitung jumlah semua pola pada masing masing isi/elemen sequences dengan menggunakan fungsi count mer tadi yang telah kita buat, dari x merupakan iterasi pada elemen sequences yang akan dihitung totalnya.

5

In [152...

```
def komplemen(x):
    return {'A':'T', 'T':'A', 'C':'G', 'G':'C'}.get(x)

def reverse_komplemen(dat):
    return map(lambda x: komplemen(x), dat)

res=reverse_komplemen(dat)
print(*res)
```

```
A C A G A A G G C C G A C T C G C C A A G G A T T G G T C G T C T G A C T A T G A C
C A G C T T A T A G C T G C C C G T T C T C G G G A C C C T A A C T A C G C A A A G
T G G T A C G C G C A G A G T C A C G T C C G T C C T T A C G T C T C G A A T G A A
G T T T G A T C A A T G A C C G T T T T T A T G T T T A A A A A A G C T A G C T G
G A A C T C A A A T A A G T A A T G G C G T G T C A G A A A A T G G C G T G G A C A
A T G G C G T G T A G G C A T T C A A A T G G C G T G C A A T G G C G T G A T G G A
G A G A T A T A A T G G C G T G A A G C A A A T G G C G T G C G A C T C C T T G C C
A A T G G C G T G A A T G G C G T G G T G T T C C A C G C A C G A G A C A A T A A T
G G C G T G G T G G T A A T G G C G T G C G T G A A A A T A A T G G C G T G G T C C
C G T G T C G G T G C A T C C C A T C G C A G C A A G A G T G A C A T A A C G C C G
C T G C C A G C A T T A A A T G G C G T A A T G G C G T G G T G A G C A A T C G A A
T G G C G T G G A T C C C A A C A A T G G C G T G C T G A A T G G C G T G T C G G C
A A T G G C G T G C A C A A T G A A C T G C G A G A T T G A G G G T G A G T A T A G
T C A G A A T A A T G G C G T G T G A C C C G A A T G G C G T G G G C G T G G A A T
T C A T C C G T C A A T G G C G T G C A T A A T G G C G T G C A T T A A T G G C G T
G T G G A C A T T T C C G T C C C A T T T C A T G T C T G A A T G G C G A A T G G C
G T G C C A A C G T G G T G C T G T T T A G A T T G C A A T C C A T G C A A T G G C
G T G C C C T T T A A T G G C G T G A G G T C C C A A A A T G G C G T G T C T A T A
G G T A A G C C C T T A C A C T G G G G A C C T C A C C T C A A C A C G C T T T C T
A T G C C T C A A A A G T T C C C G T G T G G G T C G A T A C A A T A A T T C G C A
A T G T C A C C G G C G A C G T A G T A C A G T T A C A A G T C C A G T A A G A G A
T A G A A C G A T A C A T G C T T G G G A G C A A T T C T C C C T C A T T C G C T A
G A A A A C T G T T T T A G C A T A C G T A C A T C C G C T C C G T T A C G G C T A
A T G T A A C T T G C C G C C C T G A A A A G C A T A C T C T G T G G C G C C A A C
T T T A T A A A A A A A T A C G T T C T C G C C C T A A C C C G C C T T C C T C T G
A A T T G C G T C A C G G A T C G T G A C A A T T G A C G C C G T A C C G G C C T A
C C T G A T G G A T A A A A C G T C G A G G T C G C A A A C T C A A G G T G C A T G
A C T G C C T T G T C A G G G C T C T A T C C G G T A C A C C A G C T A G G G T C A
C T C T T T A C T C T G A G C T C T A C G G C C A T G G C C A T C G T A G T G G T G
T A A C G A G G T C A T A C T A T A G T C A G A A G T G A C A G T C G T T A A T T A
C G T C G C T A G A A C T T C T C T C A A T A A G T A G A G A A T A G T G G A C T G
T T A T T T A G T T A A A T G G T C A G T T T A A G A G A A A T T G T A G C A C G G
C T T G A C G C T A C G C A G C A T C A G A T C T A A T C C T A T A T A A A A G A A
T C G A C C G A A G C T A C T A A C C G A C A T G C G A T T C C A C T A A C T T A A
A G C T A G A C G T A A C C T C G A C A T G G G G T G G A A C G T A C C G T A A C T
G T C G G A T T T C G C A C T T C T T A C G T T A T G T C G A C T G T C T T T T T A
T T G C C C G A G C T A T T G C A A G G T T C T A A G A C T G A A T T G C T G C C G
A T C G C T C G C T C A G T A T T T A G G G C A G G T G T G G C C C G T T A G C C C
A G C C T C A C C T T T C C C G C C C T A A A A T A A T A A T G C A C T G C G T C T
A G A G G C A C A G T G A T A T G A G T G T A G G A G A G A C A T C T A T T T C A A
T A T G G T T G G A G G T A T A A G A A G A A T G C G A T T C A A G C C C G A T A G
G C T C A G A G C C G G G T A T C G T C C T C G T G A A A T T C C C T T C A G G A T
A A C G G C T T A T G T C A T G C A A G G G G C G T T A T A C A A T A T G A G T G G
G T T T A T A C A A T T A T A C A A T A T A C A A T T T T G C G T C A C A C C C T T
A T A C A A T T A T A C A A T A C A C T T A T A C A A T T T T A T A C A A T T T T A
T A C A A T T G C T A C A A T C G G C A C T A T T T A T A C A A T A A T T G C C G C
A C G C A A T T A T A C A A T C G C T G C T G A C C C C A G T T A T A C A A T C G
G T T G A A G G A G T T A T A C A A T T G G C C A A T T A T A C A A T C A A T T C T
A G T T A T T T A T A C A A T C G A T G C A T C T G T T A T T T T C G T A T T C G T
T A T A C A A T A T T A T A C A A T C T G T C A A G A G A T T G G C T A T T A T A C
A A T T C C G T A T G A A T T G G T C G C T T A C T G T C T T T A T A C A A T T A T
A C A A T T T A T T T A T A C A A T C T A T T A T A C A A T G C T A T A A T G G G C
G T G T A A C G A G G C T T A T A C T T A T A C A A T T C C A C C A A G A G G C A T
A A A T T A T A A C A C T C T C T A T C G A A C A C T C T C T A C A A C A A C A C T
C T C T C G A C A C T C T C G A A A C G C T C G G A A A T T T A T A A C A C T C T C
```

A A T A C A T C A G C C G G A C A C T C T C T A C A C A C T C T C T C A A T T T A T
A C A A T C A A T C G A C T C T C G A T G C G A C A C T C T C C G C T T A A C T G C
A T C A C G A A A A C A A G A C A C T C T C T A C A C A C T C T C A C A C T C T C
G C G C A C A C T C A C A C T C T C A C T A A C A C G T A C C A G G T C A T T C T A
C A C T C T C A C A C T C T C A C T A G A T T G C G A T A C A C A C T C T C T C C C
A C A C T C T C C G A C G C A T A C T T C G T G T T T A C A C T C T C A A C A C T C
T C T A T G C A A T T C T C G G G C C T T C G A G C C G T A G T A T T C G A C T C G
T C T A A G T T A C A C T C T C C C G C T C G G C T G C C A T C C G A C A C T C T C
A G T A A T A A C A C T C T C A G C G C A C C A C A C T C T C A G G G T A A A A T A
C A C T C T A C A C T C T C G A G A C C C C G A C A C T A C A C T C T C T C A T G C
G G C T T C G C A C A C T C T C A G G A C A C T C T C T A A G C C T C C A G A C C T
A C T G T A A C A C T C T C G G A C G A A T G C G C T G C A C T A C T T G C G C T G
G C T G A T C G C T G G C G G G T G A T G A T G A G C G T C A A C C A G A T C T C C
G T A A C G A A A T G A C T T T A T G C G T C C T A C G A A T A C T G C G A G C G C
G G T T A T G T A G C G C G A G C G T G A C A T A C A G C G A A G T G G A A T T A G
G A T T T C G A G T T T A T A T T G C C T T T T T C T C T T T A A T C C T G C T G G
C T C C C A G C A G G A G G C C A C C A A A A G T G C T G A A G C G G T T A C C G C
A C G A C G C A G C T T T A C A C G A G T T T C G G G G C A T T T C G A G T C T G T
G G T A C G T C C T T A C C C T T A C A C A T G G G T C T C T A G G G A T C A T T C
T C T C T A G G T T C T G A A T T T C G G C A A G G C T C T C T C T A G A T T A G T
G A T C T C T A G A A T T G T G G T T A T C T C T A G G A G A T T C T C T A G T C A
T C T C T A G C G A A A A G T C T C T A T C T C T A G T G A G T G G C T C T C T A G
A A T G T C A A A C T A T A C A G T C A A G C C A A T T T T C G T C T C T A G C A G
A C G T C T C T A G C C A T C G C A T C T C T A G G G C A C A G C A T G T T T T T G
A A T C T C T A G T C T A G C G C G G A G C T T G A C A T G A A T C T C T A G A T G
T A A T A G A T T C T C T C T C T A G T C T C T A G T G T T C C G G T G T G T G C T
G T T T C A A T C T C T A G A T G T G T G C T A T C C A C C A C G G C T T G G A C T
C T C T A G G C C C A A A A C T C T C T C T A G T T C T C T A T C T C T A G C A A T
C T C T T C T C T A G A T C T C T A G C G T G C C C A A A A C C T C T C T A G C A A
G C C C A A A A C A G C C C T T C T C T A A T T A C G C C A C T C A A T T A C G C C
C A T A T T A C G C C G T C T A T T A C A T T A C G C C A G A T T A C A T T A C G C
C C T C T A T T A C G C C A C T A C T T T G A A T T A C G C C G A T T A C G C C A A
T T A C G C C A G C T T G C G A T T A C G C C T C G A T T A C G C C C G C A T T G T
A T T A C A T T A C G C C A A C A G T T A T A A C A A A A G T T A T A A T T A T A A
G T T A A T G T T A T A A G T T T G C G T T A T A A T T T G C C G G C C A T T A C G
C C C C A G T T A G T T A T A A A A G C A T T A C G C C C C A A T T A C G C C A A A
A G T T A T A A T A G C C A T T A C G C C C T C G A C C G T T A T A A C C A A A A C
C A T T A C G C C A A A T T A T T A C G C C C C G C T G T T A T A A C C C A T T A C
G C C T A T A A A T A G T T A T A A C A C A A A G T T A T A A A T T G T G T T A T A
A A C G G C A T C C A T A C T G G A T T A A T T A C G C C T A T A A T C C C C G G T
T A A T T A C G C C T A G C A T T A C G C C C A G C C C G A A T A T T G T T A A T T
A C G C C A G T T A T A A T G A T T A C G A T T A C G C C G C C T G A T G T T A T A
A A T G T T T T C T G A T G G T T A T T A C G C C T A T T A C G C C A G T T A T T A
C G C C T T C T A T T G C G C C G T T A T A A C G G G C T G T T A T A A A C T G A T
G T G T T C T G A T G T G T T A T A A G G C A A T A A G A C A C G G T T G C G G T C
C A G T T A C G C A G C T T G G T T A T A A G A A C T A A C A C T A C G T C T G A T
G T G C T G A T G T T A T A A A T G G G G G C C C T G A T G T A T A G G T G C T G A
T G T C C C G C T C T G A T G T A T C C C T G A T G T C T G A T G T T G T T A A T A
C C A G T G T A A T T G A G A C G G G C C G C C G A G A A G G G A T T T A G A G T G
C A C T A C C T G A T C T G A T G T G G C T G A T G T T G T A T G A A A C G T T G C
T G A T G T C A T G C A A T T C T G A T G T C C T A A G T C T G A T G T G A A C T A
A A G G A C T G A T G T G A A G A C T G T T G G G C G T G T A A C G G G C G A T T G
A G A C T A C C G G G G G T C T C T G A T G T A T G G T A G C T C G C G C T G A T G
T C C T G A T G T C G G C A T C T G G G A A A T C T G A T G T G C G G T C C C G G T
T G A C T G T G C C T A T T C C A G A A A C G G G G C G T T C A C G A G C G G C T T
A C A C T A A T T A G A G T T G T A A G G C T G G A C G T T C T C G T G T G C G T A
A A C T A T A C C C A T A T T C C T T C T A G A G C A G G T C G A T A T T A C A T G
T T G T A A A G G G G C A G T A C T G A A C G A T G T A T T T G T C T T A T T C T G
C A C T G C A G C G G T T A T A T T C T G C A T G A G C T A A C T G G C A T T T T A
A A A A G A T T C T T G G T T T T A T T C T G C A T T C T G C A A G T G A A T T T A
T T C T G C A T C C C G C A A T G G C T A T T C T G C A A T T C T G C A C C T A G C
G G T A G C G G G C A C T C A G C G A G A G G G C G T A T T C T G C A A T T C T G C
A G G G T T A T C A C G A G G G A T G T G A A A T G G C C A C C A T C T A T T C T G
C A T C T G C A A T A T T C T G C A G C C C A T T T A T A T T C T G C A A T A A G G

G T T A T T T A T T C T G C A T T A G G G A C A T T G T G A C C T T C A C T A T T C
 T G C A A A C A A G A T T G T A T T C T G C A A C A T T G A C G G G A T T G G G A C
 T A T T C T G C A A A A T T T T T C A T G A T A T T C T G C A A G C T C C T T A C T
 C T G G T A T T C T G C A G C A G G G A G G G A G T C G T G A C T T A A A A A A G C
 T T C T A T T C T G C A T T C T G C A C A A C C A A A T A G C A A T C T T T A T T C
 T G C A T G C A A A T A T T C T G C A T T A C C A G T A T T C T G C A T G C A A T T
 C T G C A T T A T T C T G C A A T A G G T A G G G T T T T A A T G T G C A G T C T T
 T A G T A C C G T T G G C G G C A C T A C C T T C T C A T C G T T G G C T G A T
 G T A T G T C A T A T G A C A C C C G T C T G A G C A A A C A T G T G G T T G T G A
 A G G C G G C G G T A A T A A T T T A T G C T A A C C A C G A A A T G C G T A G A A
 C T A C T G G T A C C A A T G A G T G G A G C C C A C G A C T G G G C G G A C A G A
 G G A T A C T G C A G C C C G A G G T G A T G C C G G G G C A A A G C T G T C T A T
 C C C C C C T C A A C T G G A G C T T A C G C C C A A T G A A G C G G A C G G A A A
 G C T G C T T A G C C A T A C C G A T C G A A C C T G T T C A T A T C C T A A C C A
 G A A A G T T C G A C G T G A C A A A A C G T C G A A G A T C G C T C T A T T C C G
 A C T T C G G A G G T C G C T A T A A C A G G T C A A C C T T T T T T C A A C C T T
 T T T A C C C C C A A A C C T T T T T T C T T T T G C G G G C C C A A T G T G G C C
 C C T G T A T T A A C C T T T T T T G G T C A A C C T T T T T C G A A T C T T T C G
 A A C C T T T T T C A G A A C C C T T G T T A A T A A C C T T T T T G C T A C C C G
 C T G A C T C T C A A C C T T T T T T T T A A C C T T T T T A A C C T T T T T G C
 A C C G A A A C C T T T T T A C C T T T T T C T A A C C T T T T T A A C C T T T T T
 G A A A A C C T T A A C C T T T T T T T T C G G T G A C G C C C A C G A A A C C T T
 T T T T A T A A A C C T T T T T T G A T C G T T T C G C C G T A A G A C T C T C T A
 A C C T T T T T G C A C G A T T C G A A G A A A C C T T T T T C T T A A C C T T T T
 T T T T T T C G C G T G G T G A G T C C T T C T G T A C A G A C C G T G A A A T C G
 C A A T T T C A A A C C T T T T T T G A G G A G G G T G T A A A C C T T T T T A C C
 T T T T T C T T A G C C A A T C T C G C C G T G C A C A G T A T A A C C T T T T T A
 T G A G T C G C G C A A T C G T C A A C C T T T T T T A C T A C T G A T A C A A A
 C C T T C T G T T C C T C T T T C A G A G G C T T G T T G T A G G T A C T G T T C C
 T C C T C C G A C C T G T T C C T A A G T C C G A C A A G T C T G T T C C T C C C T
 G C T G T T C C T T C C T G A C A A G T C C G A C C T G T T C T G T T C C T G A C A
 A C T G T T C C T G T C C T G T T C C T G C T T T C C G A C A A G T C C C T G T T C
 C T T C C T G T C C G A C A A G T C T C C T G C T C C T G C T G T T C C T T C C G A
 C A A G T C C G A C A A G T C C T C C T G C T C C T T C C T A C A A G C T G T T C T
 C C T G C T G T C C G A T C C T G C T G C T T C T C C T G C T G A C A A G T C C G A
 C A T C C T G C T C C T G C T T T C C T A C A A C T G T T C C T C C T C T C C T C C
 T G C T C C T T C C T G C T T T C C T C T G T T C C T C T G T T C C T C T G T T C C
 T T C C T G C T T C C T G C T T C C T G C T T C C T G C T T C C T G C T T C C T G C
 T T C C T G T C T C C T G T C C T G C T T C C T G C T T C C T G C T T C C T G
 T C C T G C T T C C T G C T T C C T G C T T C C T G C T G T T A G T A G T T A G T A
 G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A
 G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A
 G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A
 G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A
 G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A G T T A G T A

digunakan reverse komplement untuk mendapatkan komplement terbalik dan komplement
 REVERSE dari 'dat' digunakan untuk mendapatkan iterator terbalik dari sequence yang diberikan
 pertama kita buat fungsi komplement dulu yang isinya dictionary yang didalamnya terdapat key
 dan value yang sesuai untuk menjawab pertanyaan no 5 di jurnal. setelah itu kita gunakan code
 get(x) untuk mendapatkan nilai dari valuenya saja.

kemudian kita buat fungsi kedua yang dimana digunakan untuk menghitung value dari variabel
 dat dengan cara menggunakan fungsi komplement yang sudah kita buat sebelumnya

In [102...

```

import math

def aktivasi(x):
    return 1/(1+math.exp(-x))

def WTi(W,i):

```

```

    return list(map(lambda w: w[i], W))
def WT(W):
    return list(map(lambda i: WTi(W,i), range(0,len(W[0]))))
def XW(X,W):
    return map(lambda w:r(lambda x,y:x+y, map(lambda xx,ww:xx*ww, X,w)), WT(W))
def input_to_hidden(X,W):
    return map(lambda x: aktivasi(x), XW(X,W))
def feed_forward(X,W,M):
    return input_to_hidden(input_to_hidden(X,W),M)

```

In [103...

```

X=[9,10,-4]
W=[[0.5,0.4],[0.3,0.7],[0.25,0.9]]
M=[[0.34],[0.45]]
print(list(feed_forward(X,W,M)))

```

[0.6876336740661236]

code diatas adalah fungsi fungsi yang dibuat untuk digunakan dalam menghitung hasil feed_forward dan code selanjutnya adalah kita menginisialisasi nilai nilai yang akan digunakan sesuai soal seperti input x, lalu input weight dan nilai M nya.

untuk lebih dapat memahami fungsi yang telah dibuat diatas. kita dapat coba mengeluarkan hasil atau output yang didapat dari masing masing fungsinya. seperti dibawah ini :

In [130...

WTi(W,0)

Out[130...

[0.5, 0.3, 0.25]

fungsi WTi diatas adalah digunakan untuk mendapatkan value nilai dari W berdasarkan indeksny dari masing masing list atau WTi digunakan untuk men transpose dengan inputan W dan return ke I lalu mapping W menjadi satu dimensi

In [131...

WT(W)

Out[131...

[[0.5, 0.3, 0.25], [0.4, 0.7, 0.9]]

selanjutnya fungsi WT ini adalah result atau hasil berupa list dari memisahkan value berdasarkan indeks pada fungsi WTi tadi atau WT bisa dibilang untuk melakukan perkalian x1 dan x2 dengan dilakukan mapping untuk setiap elemen nya

In [135...

list(XW(X,W))

Out[135...

[6.5, 7.0]

Untuk fungsi XW ini adalah fungsi yang berisikan perhitungan untuk mengkalikan silang nilai nilai pada value WT.

In [134...

print(*input_to_hidden(X,W))

0.998498817743263 0.9990889488055994

input to hidden ini adalah fungsi yang digunakan untuk menghitung hasil dari XW yang telah diproses dengan fungsi aktivasi(x), kemudian didapatkan nilainya.

In [154...

```
print(list(feed_forward(X,W,M)))
```

```
[0.6876336740661236]
```

lalu terakhir adalah fungsi feed forward untuk mendapat hasil terakhir yang kita cari. Pertama adalah memprosesnya nilai pada value M ke perhitungan aktivasi(x), kemudian setelah didapatkan hasilnya di proses lagi dengan aktivasi(x) menggunakan fungsi input_to_hidden tadi lalu didapatkanlah hasil akhirnya

In []: