**CSGE601020** **Dasar-Dasar Pemrograman 1**

Nama: Kelas:

NPM: No bangku:

**UAS – 120 minutes – 17 Desember 2016 -- Dengan Catatan Tulisan Tangan**

***Baca setiap soal dengan cermat dan tulis jawaban Anda dengan jelas pada tempat yang disediakan. Ada 11 soal. Nilai total adalah 110 poin.*** *Salah satu soal (terserah Anda) adalah soal bonus.*

1. [5+5] Fragmen program berikut ini dimaksudkan untuk membangun sebuah list L yang berisi 100000 bilangan gasal, tetapi kurang efisien:

n = 100000

L = []

for i in range(n):

L = L + [2\*i + 1]

Perbaiki dengan mempergunakan:

(a) method append

(b) list comprehension

**Jawab:**

(a)

(b)

1. [10] Cari output yang dicetak pada layar oleh program ini.

def swap1(param1, param2):

param1, param2 = param2, param1

return (param1,param2)

def swap2(param, i, k):

param[i], param[k] = param[k], param[i]

def main():

numList = [4,2,5,1,3]

(numList[0],numList[3]) = swap1(numList[0],numList[3])

print("After swap1, numList:", numList)

swap2(numList, 4, 2)

print("After swap2, numList:", numList)

main()

**Jawab:**

1. [10] Cari output yang dicetak pada layar oleh program ini.

lst = [2.1, 4, "string", 8, 1]

a = 0

for i in lst:

try:

a += i

except TypeError:

print('Type Error')

break

except:

print('Error')

pass

finally:

print("i: " + str(i))

print("a: " + str(a))

**Jawab:**

1. [10] Buatlah sebuah fungsi **hitung** yang menerima argumen berupa sebuah string dan mengembalikan nilai berupa list of tuple. Tuple yang ada pada list kembalian ini adalah pasangan dari kata dan jumlah kemunculan kata tersebut dalam string yang menjadi argumen. Pergunakan **dictionary**. Semua karakter dalam **string.punctuation** (yaitu '**!"#$%&\'()\*+,-./:;<=>?@[\\]^\_`{|}~**' ) diperlakukan sebagai separator**.** Anda diperbolehkan menambahkan fungsi lain untuk membantu operasi dalam fungsi **hitung**.

Contoh:

>>> hitung("Saya jalan-jalan ke pinggir jalan,.")

[('pinggir', 1), ('ke', 1), ('Saya', 1), ('jalan', 3)]

**Jawab:**

1. [10]

|  |
| --- |
| def mystery(a,b):  for i in range(1, a\*b):  if ((i % a == 0) and (i % b == 0)):  print(i)  break |

Apa output fungsi di atas jika dimasukkan nilai a=12 dan nilai b=30?

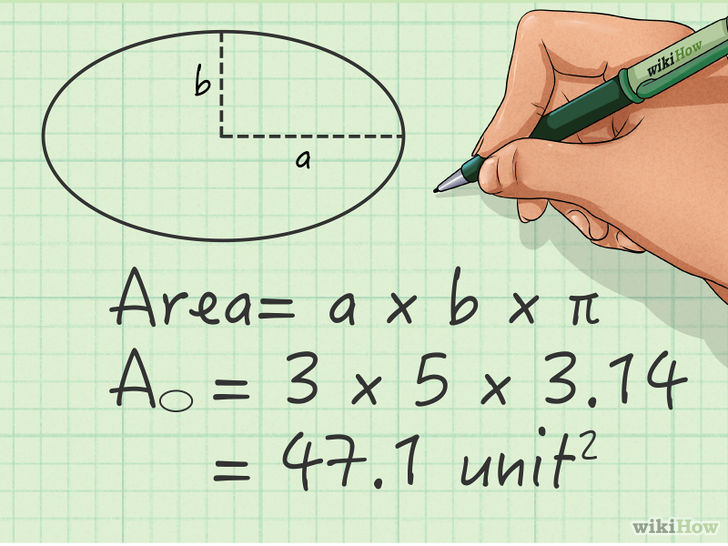
Apa yang dilakukan fungsi ini secara umum?

**Jawab:**

**6. [10]** Lengkapi fungsi **sumAll()** di bawah ini untuk menghitung jumlah semua digit yang ada di dalam suatu file teks. Misal, jika isi file **‘input.txt’** adalah **'1 saya 2 belajar 3 ddp1'**, maka pemanggilan fungsi sumAll(‘input.txt’) akan mengembalikan nilai **7**, yaitu **1+2+3+1**.

|  |
| --- |
| def sumAll(filename):  infile = open(filename,'r')  sums = 0  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  infile.close()  return sums |

7. [10] **Elips** adalah sebuah bentuk di bidang datar, seperti yang terlihat pada gambar di bawah. Sumbu **a** disebut dengan **sumbu semi-major**, dan sumbu **b** disebut dengan sumbu **semi-minor**.



Luas dari elips ditentukan oleh persamaan berikut:

**Luas = π x *a* x *b***

Misalkan, Anda sudah mempunyai definisi dari kelas Elips sebagai berikut:

|  |
| --- |
| import math  class Elips:  def \_\_init\_\_(self, a, b):  self.\_\_a = a  self.\_\_b = b  def getLuas(self):  return math.pi \* self.\_\_a \* self.\_\_b |

Tugas Anda adalah membuat definisi kelas **Lingkaran** yang merupakan **bentuk yang lebih spesifik** dari **Elips,** dimana sumbu semi-major dan semi-minor mempunyai panjang yang sama. Objek dari kelas Lingkaran mempunyai atribut **radius** dan mempunyai method **getLuas()** dan **getRadius()**. Pergunakan **inheritance** semaksimal mungkin. Jangan buat method baru yang tidak dibutuhkan.

|  |
| --- |
| class Lingkaran(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_): |

1. [10] Buatlah sebuah fungsi rekursif **pairStar** yang menyisipkan karakter bintang (\*) diantara karakter yang identik. Fungsi ini menerima argumen sebuah string dan mengembalikan sebuah string. Berikut ini adalah contohnya:

pairStar("hello") → "hel\*lo"  
pairStar("xxyy") → "x\*xy\*y"  
pairStar("aaaa") → "a\*a\*a\*a"

pairStar("UI") → "UI"

**Jawab:**

|  |
| --- |
|  |

9. [10] Cari semua output yang dicetak oleh program ini:

**class A(object):**

**def \_\_init\_\_(self, i = 10):**

**print("calling init in class A")**

**self.i = i**

**def m(self):**

**print("calling m in A")**

**self.i //= 2**

**class B(A):**

**def \_\_init\_\_(self, j = 100, i = 50):**

**print("calling init in class B")**

**A.\_\_init\_\_(self)**

**self.j = j**

**def m(self):**

**super().m()**

**print("calling m in B")**

**self.j \*= 5**

**def main():**

**b = B() #Line 1**

**b.m() #Line 2**

**print(b.i) #Line 3**

**print(b.j) #Line 4**

**print(isinstance(b,A)) #Line 5**

**main()**

**Jawab:**

#Line 1:

#Line 2:

#Line 3:

#Line 4:

#Line 5:

1. [10] Buat sebuah fungsi rekursif **countAll** yang dapat dipanggil dengan pola berikut:

countAll( elem, L )

yang me-**return** jumlah dari semua kemunculan integer **elem** dalam list **L.**  Integer **elem** dapat muncul pada sembarang level dari struktur list **L**.

Contoh:

>>> lst = [3,[7,[3,9]],[3,[-23,3]]]

>>> countAll(3,lst)

4

>>> countAll(-8,[])

0

>>> countAll(25, [1,25,30,[25]])

2

>>> countAll(1,[[[1]],1,1,[1],1,3,[]])

5

**Jawab:**

def countAll(elem, L):

1. [10] Cari output yang dicetak oleh program berikut ini, yang dilaksanakan sebagai *main module*. Apa sebenarnya yang dilakukan oleh program ini?

def m(s):

if len(s) == 0:

return ['']

strings = []

for i in range(len(s)):

subStrings = m(s[:i]+s[i+1:])

for st in subStrings:

strings += [s[i] + st]

return strings

def main():

n=3

ALPHABET = 'abcdefghijklmnopqrstuvwxyz'

st = ALPHABET[:n]

strings = m(st)

for astr in strings:

print(astr)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Jawab:**

===☺Never Give Up☺===