

1.	what does the following code do? (เลือกคำตอบที่คิดว่าถูกต้อง)
	def a(b, c, d): pass type(a(1,2,3))
	<input type="checkbox"/> defines a list and initializes it
	<input type="checkbox"/> defines a function, which does nothing
	<input type="checkbox"/> defines a function, which passes its parameters through
	<input type="checkbox"/> defines an empty class
	อธิบายว่า เข้าใจว่ามันคืออะไร

คือการสร้างฟังก์ชันแต่ไม่มีโครงสร้างภายใน จึงใช้ **pass** เพื่อเอาไว้ทดสอบ โดยเรียกใช้ฟังก์ชัน **type** เพื่อ **return** ประเภทของตัวแปรโดยในที่นี้จะ **return NoneType** เนื่องจากภายในฟังก์ชันไม่มีการ **return** ค่า

2.	what gets printed?
	>>> type(1/2)
	Result = _____
	อธิบาย _____

Result = float

อธิบาย เนื่องจาก 1/2 ได้ 0.5 ซึ่งเป็นทศนิยมจึงทำการ **return** ค่า **float** ออกมา

3.	what gets printed?
	>>> type(1j)
	Result = _____
	อธิบาย _____

Result = complex

อธิบาย ใน **Python** ถ้าอยากสร้างจำนวนเชิงซ้อน จะใช้ **J** หรือ **j** ต่อหลังเลข จึง **return** เป็น **complex**

4.	what gets printed?	5.	what gets printed?
	>>> type((1,2))		>>> type([1,2])
	Result = _____		Result = _____
จากผลการทดลองข้อ 4 และ 5 เหมือนหรือต่างกันอย่างไร _____			

Result ข้อ 4 = tuple

Result ข้อ 5 = list

อธิบาย เมื่อต้องการประกาศสมาชิกแบบ **tuple** จะใช้ **()** แต่ **list** จะใช้ **[]** โดยข้อแตกต่างอีกอย่างคือ หลังจาก **tuple** สร้างไปแล้วจะไม่สามารถแก้ไขค่าได้แต่ **list** สามารถทำได้

6.	what gets printed?
	>>> def f():pass >>> type(f())
	Result = _____

Result = NoneType

อธิบาย NoneType จะไม่มีการส่ง และ รับพารามิเตอร์

7.	what gets printed?
	>>> a = [1,2,3,None,(),[],] >>> print len(a)
	Result = _____

Result = 6

อธิบาย คำสั่ง len เป็นการหาจำนวนค่าในตัวแปรที่อยู่ใน ()

8.	what gets printed?
	>>> type(lambda:None)
	Result = _____
	Lambda คือ _____

Result = function

อธิบาย Lambda คือการสร้างฟังก์ชันที่ไม่มีพารามิเตอร์

9.	>>> nums = set([1,1,2,3,3,3,4]) >>> print(len(nums))
	Result = _____
	ทำไมถึงเป็นเช่นนั้น _____

Result = 4

อธิบาย set จะตัดค่าที่ซ้ำออกและใช้ len นับค่าที่เหลืออยู่

10.	>>> d = lambda p: p * 2 >>> t = lambda p: p * 3 >>> x = 2 >>> x = d(x) # x = _____ = _____ >>> x = t(x) # x = _____ = _____ >>> x = d(x) # x = _____ = _____ >>> print(x)
	Result = _____

$x = d(x) \# x = 2 * 2 = 4$

$x = t(x) \# x = 4 * 3 = 12$

$x = d(x) \# x = 12 * 2 = 24$

Result = 24

11.	<pre>>>>def test(x,y,z): if x or y and z: print("yes") else: print("no") >>>test(1,0,0)</pre>	List เงื่อนไขที่ทำให้ test แสดง False _____ _____ _____ _____

เงื่อนไขที่ทำให้เป็น false

x = 1 , y = 0 , z = 0 = false

x = 0, y = 1 , z = 0 = false

12.	<pre>>>> x = 4.5 >>> y = 2 >>> print(x//y)</pre>
	Result = _____
	// คือ _____

Result = 2.0

// คือ การหารและตัดเศษทิ้ง

13.	<pre>>>> x = 4.5 >>> y = 2 >>> print(x/y)</pre>
	Result = _____
	/ คือ _____

Result = 2.25

/ คือ การหาร

14.	If PYTHONPATH is set in the environment , What option that not include in directory searched for modules? A) PYTHONPATH directory B) current directory C) home directory D) installation dependent default path
	Enter Your choice = _____ Explain, Why do you select this answer? : _____

เลือก C

การทำงานจะเริ่มโดยทำการค้นหา current directory แล้วไปที่ pythonpath directory และจบที่ dependent default path

15.	<pre> x = True y = False z = False if not x or y: print(1) elif not x or not y and z: print(2) elif not x or y or not y and x: print(3) else: print(4) </pre>	<div>ใส่วงเล็บแสดงลำดับการทำงานของ if พร้อมแสดง Logic ที่ได้</div> <hr/> <div>ใส่วงเล็บแสดงลำดับการทำงานของ elif พร้อมแสดง Logic ที่ได้</div> <hr/> <div>ใส่วงเล็บแสดงลำดับการทำงานของ elif พร้อมแสดง Logic ที่ได้</div> <hr/>
	Result = ____	

x = True

y = False

z = False

If not x or y = if False or False = False

elif not x or not y and z = elif False or True and False = False

elif not x or y or not y and x = elif False or False or True and True = True

Result = 3

16.	<pre> daysOfWeek = ['Monday','Tuesday','Wednesday', 'Thursday','Friday','Saturday','Sunday'] months=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'] print("DAYS: %s" % (daysOfWeek)) print("MONTHS %s" % (months)) </pre>
	อธิบายการทำงาน <hr/> <hr/> <hr/> <hr/>

อธิบาย สองบรรทัดแรกจะเป็นการประกาศตัวแปรแบบ list และบรรทัดที่ 3,4 ทำการแสดงข้อความโดยมีการใช้ format %s เพื่อแสดงข้อมูลแบบ String โดยนำตัวแปร list มาแสดงแต่ไม่ได้กำหนดลำดับของค่า จึงแสดงค่าทั้งหมดออกมา

17.	<pre> counter = 1 def doLotsOfStuff(): global counter for i in (1, 2, 3): counter += i doLotsOfStuff() print(counter) </pre>
	Result = ____

Result = 7

18.	<code>print("\x48\x49")</code>
	Result = _____
	อธิบายว่า <code>x48</code> มันคืออะไร ? _____

Result = HI!

อธิบาย `x48` คือเลขฐาน16 ในรูปแบบของ python

19.	<code>print(str(0xA) + ' '+str(0xa))</code>
	Result = _____
	อธิบายว่า <code>0xA</code> มันคืออะไร ? _____

Result = 10 10

อธิบาย `0xA` คือเลขฐาน 16 โดยใช้ `str` เพื่อแปลงให้ `0xA` เป็น String ในการแสดงผล

20	<pre>>>>kvps = {"user":"bill", "password":"hillary"} >>>print(_____)</pre> <p>Bill Hillary</p>
----	--

`print(kvps["user"] + " " + kvps["password"])`

ออกแบบโปรแกรมคำนวณรับค่าตัวต้านทาน 2 ตัว พร้อมระดับแรงดันอินพุต (ด้วย input และ output GUI)
ในวงจรอนุกรมและขนาน แสดงค่า R_T , R_1 , R_2 , V_T , V_1 , V_2 , I_T , I_1 , I_2 (R หน่วยเป็น Ω , $K\Omega$, $M\Omega$ ('/u03A9'))
พร้อมแสดงค่าที่เหมือนและที่แตกต่างในแต่ละจุด
กรณีเหมือนแสดงในบรรทัดเดียวกัน
 $V_T = V_1, V_2 = 5 \text{ V.}$
กรณีต่างแสดงผลต่างบรรทัด
 $R_T = 3.5 \text{ K}\Omega$
 $R_1 = 1.5 \text{ K}\Omega$
 $R_2 = 2 \text{ K}\Omega$

Code

import tkinter as tk เรียกใช้ library

H = 500 กำหนดความสูง

W = 800 กำหนดความกว้าง

root = tk.Tk() สร้าง object

canvas = tk.Canvas(root, height=H, width=W) กำหนดรูปแบบฟอร์ม

canvas.pack() สร้างฟอร์ม

labelR1 = tk.Label(text='R1',font=('Courier',10)) สร้าง label

labelR1.place(relx=0.05,rely=0.05) กำหนดตำแหน่ง label

labelR2 = tk.Label(text='R2',font=('Courier',10)) สร้าง label

labelR2.place(relx=0.05,rely=0.12) กำหนดตำแหน่ง label

label1v = tk.Label(text='Volt1',font=('Courier',10)) สร้าง label

label1v.place(relx=0.04,rely=0.19) กำหนดตำแหน่ง label

R1_input = tk.Entry(font=('Courier', 10)) สร้าง label

R1_input.place(relx=0.1,rely=0.05,relwidth=0.5, relheight=0.05) กำหนดตำแหน่ง textbox

R2_input = tk.Entry(font=('Courier', 10)) สร้าง textbox

R2_input.place(relx=0.1,rely=0.12,relwidth=0.5, relheight=0.05) กำหนดตำแหน่ง textbox

`v1_input = tk.Entry(font=('Courier', 10))` สร้าง textbox

`v1_input.place(relx=0.1, rely=0.19, relwidth=0.5, relheight=0.05)` กำหนดตำแหน่ง textbox

`button = tk.Button(text='ตกลง',`

`font=40,command=lambda:show(R1_input.get(),R2_input.get(),v1_input.get()))` กำหนดค่า button

`button.place(rely=0.05,relx=0.7, relwidth=0.2, relheight=0.1)` กำหนดตำแหน่งของ button

`lower_frame = tk.Frame(root, bg= '#000000', bd=1)` กำหนดค่าของ box แสดงผล

`lower_frame.place(relx=0.5, rely=0.35, relwidth=0.75, relheight=0.6, anchor='n')` กำหนดตำแหน่งของ box

`label = tk.Label(lower_frame, font=('Courier', 10), anchor='nw', justify='left')` กำหนดตำแหน่งของข้อมูล

`label.place(relwidth=1, relheight=1)` วางข้อความ

`def show(R1,R2,V):` สร้าง fuction def

`s = "อนุกรม\n"` เก็บค่า String

`r1 = float(R1) * 1000` แปลงค่าตัวด้านทาน

`r2 = float(R2) * 1000`

`rt = (r1+r2)/1000` หาค่าต้านทานรวม

`it = float(V) / rt` หากระแสรวม

`i1=i2=it` กระแส1

`v1 = (it*r1)/1000` แรงดัน1

`v2 = (it*r2)/1000` แรงดัน2

`if(v1 == v2):` ถ้าแรงดันเท่ากันแสดงค่าบรรทัดเดียวกัน

`s = s + "VT = V1,V2 = " + str(v2)+"V\n"`

`else:` ถ้าไม่เท่ากันให้แสดงข้อความแยกบรรทัด

`s = s + "VT = "+str(it*rt)+ "\nV1 = " +str(v1) + "\nV2 = " + str(v2)+"\n"`

`if(r1 == r2):` ถ้าค่าต้านทานเท่ากันแสดงค่าบรรทัดเดียวกัน

`s = s + "RT = R1,R2 = "+ str(r2/1000)+"\n"`

else: ถ้าไม่เท่ากันให้แสดงข้อความแยกบรรทัด

```
s = s + "RT = "+str(rt) + "\nR1 = " + str(r1/1000) + "\nR2 = "+ str(r2/1000)+"\n"
```

if(i1 == i2): ถ้ากระแสเท่ากันแสดงค่าบรรทัดเดียวกัน

```
s = s + "IT = I1,I2 = " + str(i1)+"\n"
```

else: ถ้าไม่เท่ากันให้แสดงข้อความแยกบรรทัด

```
s = s + "IT = "+str(it)+"\nI1 = "+str(i1)+"\nI2 = " + str(i2)+"\n"
```

s = s + "ขนาน\n" แสดงข้อความ

r1 = float(R1) * 1000 หาค่าตัวต้านทาน

r2 = float(R2) * 1000 หาค่าตัวต้านทาน

rt = ((r1*r2)/(r1+r2)) หาค่าต้านทานรวม

rt = rt*(10**-3)

it=float(V)/(rt*1000) หาค่ากระแส

v1=v2=float(V) หาค่าแรงดัน

i1=v1/r1 หาค่ากระแส

i2=v2/r2 หาค่ากระแส

if(v1 == v2): ถ้าแรงดันเท่ากันแสดงข้อความรวมกัน

```
s = s + "VT = V1,V2 = " + str(v2)+"V\n"
```

else: ถ้าแรงดันไม่เท่ากันให้แสดงข้อความแยกบรรทัด

```
s = s + "VT = "+str(it*rt)+ "\nV1 = " +str(v1) + "\nV2 = " + str(v2)+"\n"
```

if(r1 == r2): ถ้าค่าต้านทานเท่ากันแสดงข้อความรวมกัน

```
s = s + "RT = R1,R2 = " + str(r2/1000)+"\n"
```

else: ถ้าค่าต้านทานไม่เท่ากันให้แสดงข้อความแยกบรรทัด

```
s = s + "RT = "+str(rt) + "\nR1 = " + str(r1/1000) + "\nR2 = "+ str(r2/1000)+"\n"
```

if(i1 == i2): ถ้าค่ากระแสเท่ากันแสดงข้อความรวมกัน

```
s = s + "IT = I1,I2 = " + str(i1)+"\n"
```

else: ถ้าค่ากระแสไม่เท่ากันให้แสดงข้อความแยกบรรทัด


```
s = s + "IT = "+str(it)+"\nI1 = "+str(i1)+"\nI2 = " + str(i2)+"\n"
```

label.config(text=s) แสดงข้อความ

root.mainloop() ให้ทำงานเป็น loop

ผลลัพธ์

