



คณะวิศวกรรมศาสตร์
มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต

การทดสอบความรู้ขั้นปริญญาตรี
วิชา ว.พ.101 การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น
วันศุกร์ที่ 14 ธันวาคม พ.ศ. 2561

ปลายภาคการศึกษาที่ 1 ปีการศึกษา 2561

เวลา 13:00 - 16:00 น.

ตอนที่:

☐ 810001 จ. 09:30 - 12:30 รศ. ดร. วีรชัย โอโณทัยไพบูลย์

☐ 760001 อ. 09:30 - 12:30 อ. วชิรา พรหมสาขา ณ สกลนคร

ชื่อ-นามสกุล: _____ เลขทะเบียน: _____ เลขที่นั่งสอบ: _____

คำสั่ง: ให้นักศึกษาอ่านคำสั่งนี้โดยละเอียดก่อนลงมือทำข้อสอบ

1. ให้นักศึกษาเขียนชื่อ เลขทะเบียน เลขที่นั่งสอบให้ชัดเจน พร้อมระบุวิชาและหมายเลขตอนให้ถูกต้อง
ทั้งบนหน้ากระดาษคำถามและกระดาษคำตอบทุกแผ่น มิฉะนั้นจะถูกหัก 5 คะแนน
2. ข้อสอบมี 2 ส่วนคือ กระดาษคำถามจำนวน 10 หน้า (รวมหน้าปก) และกระดาษคำตอบจำนวน 6 หน้า รวมทั้งหมด 16 หน้า
และให้ตรวจข้อสอบให้ครบทุกหน้าก่อนลงมือทำ
3. ให้เขียนคำตอบลงในกระดาษคำตอบ
4. ข้อสอบเป็นแบบเติมคำ 15 ข้อ ข้อละ 1 คะแนน เขียนโปรแกรม 5 ข้อ แบ่งเป็น ข้อละ 5 คะแนน 4 ข้อ และข้อละ 10 คะแนน 1 ข้อ
รวม 20 ข้อ คะแนนเต็ม 45 คะแนน
5. ข้อสอบแบบเติมคำ ให้นักศึกษาเขียนคำตอบเท่าที่โจทย์ถามเท่านั้น ลงในกระดาษคำตอบที่จัดให้ ตอบถูกต้องข้อละ 1 คะแนน
ตอบผิดหรือไม่ตอบ จะไม่ได้หรือเสียคะแนน คำตอบที่อ่านไม่ออกจะไม่ได้รับการตรวจ
6. ห้ามนำเอกสารทุกชนิด และอุปกรณ์อิเล็กทรอนิกส์ทุกประเภทเข้าห้องสอบ
7. ห้ามนักศึกษายืมสิ่งของใด ๆ ทุกชนิด
8. ห้ามนำข้อสอบและกระดาษคำตอบออกจากห้องสอบโดยเด็ดขาด ให้ส่งคืนทั้งชุด

หมายเหตุ

1. หากไม่ระบุให้ โปรแกรมหมายถึงโปรแกรมภาษาไพธอน
2. หากไม่ระบุ ให้สมมติว่าข้อมูลนำเข้า (input) ของโปรแกรมหรือฟังก์ชันมีความถูกต้อง ไม่ต้องตรวจสอบ
3. ในกรณีที่โจทย์ระบุเฉพาะส่วนหนึ่งของโปรแกรม ให้นักศึกษาพิจารณาเฉพาะส่วนโปรแกรมที่ให้เท่านั้น
โดยถือว่าโปรแกรมประกอบด้วยส่วนอื่นที่จำเป็นต่อการทำงานของโปรแกรม
4. ผู้ใช้ หมายถึง ผู้รันโปรแกรม

นักศึกษาโปรดตรวจสอบว่าไม่ได้นำเอกสารหรือสิ่งที่เกี่ยวข้องนอกจากที่ระบุไว้เข้าห้องสอบ หากตรวจพบเอกสารหรือสิ่งที่เกี่ยวข้อง
อยู่ในการครอบครองของท่าน ท่านจะถูกพักการเรียน 1 ปีการศึกษา และปรับตกวิชานั้นทันที

ส่วนที่ 1. เติมคำ (15 คะแนน)

1. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
def myfn(t):  
    return t[1], t[0]  
  
list1 = [[1, 2], [3, 4], [5, 6], [7, 8]]  
list2 = []  
for item in list1:  
    list2.append(myfn(item))  
print(list2)
```

2. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
list1 = []  
list1.append(4)  
list1.extend(list(range(6,8)))  
list1 += list1  
list1.remove(6)  
list1.clear()  
list1.insert(0,"10")  
list1.append(list1.count(10))  
list1 = [int(x) for x in list1]  
list1 *= 2  
print(sorted(list1))
```

3. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
sum = 0  
for x in range(1, 4, 2):  
    for y in range(2, 5, 2):  
        for z in range(1, 2, 1):  
            sum += (x+y) * z  
print(sum)
```

4. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
data = [6, 4, 9, 3]  
i = 3  
while i > 0:  
    if data[i] < data[i-1]:  
        data[i], data[i-1] = data[i-1], data[i]  
    i -= 1  
print(data)
```

5. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
data = [-1.5, 9, ("engr", (8, "python"), "cn101")]  
print(data[2][1][1][3])
```

6. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
x = 1

def main():
    x = 2
    print(a(x))

def a(w):
    global x
    x += 2
    return b() + 1, c(x)

def b(x=2):
    return c(x) * 2

def c(y=3):
    if y > x:
        return y ** 3
    else:
        return y ** 2

main()
```

7. จงเขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
t = 0
n = 2561
while n > 0:
    t *= 10
    t += n % 10
    n //= 10

print(t)
```

8. จงเขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
def up():
    global x
    x += 1

def down():
    global x
    x -= 1

x = 0
n = 10
while n > 0:
    if n % 2 == 0:
        up()
    else:
        down()
    n //= 2
    print(x, end='')
```

9. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
quote = "Happiness is a choice."  
list1 = sorted([len(word) for word in quote.split()])  
print(list1)
```

10. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
x = y = []  
for t in range(-2, 0, 1):  
    x.append(t)  
for t in range(0, 4, 2):  
    y.append(t)  
print(x)
```

11. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
n = 0  
sum = 0  
data = [22, 7, 24, 59, 10, 9, -1]  
for x in data:  
    while x >= 0:  
        if sum < 100:  
            n += 1  
            sum += x  
        break  
print(n, sum)
```

12. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
def myfn(t):  
    return t[0][1:2]  
  
data = [("cat", 2), ("dog", 3), ("ant", 4), ("fish", 1)]  
data.sort(key=myfn)  
for item in data:  
    print(item[0], end=" ")
```

13. จงเขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
t = 0  
n = 2561  
for c in str(n):  
    t += int(c)  
  
print(t)
```

14. เขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
data = [(1, 2), (3, 4), (5, 6), (7, 8)]  
for item in data:  
    x = lambda a, b: sum(a) + sum(b)  
    print(x(item[0], item[1]), end=" ")
```

15. จงเขียนข้อมูลส่งออกของโปรแกรมต่อไปนี้

```
def mystery(s):  
    s = s.lower()  
    n = 0  
    for c in range(ord('a'), ord('z') + 1):  
        if chr(c) in s:  
            n += 1  
    return n
```

```
print(mystery('The quick brown fox jumps over the lazy dog'))
```

ส่วนที่ 2. โปรแกรม (30 คะแนน)

16. [5 คะแนน] จงเขียนโปรแกรมเพื่อรับค่าช่วงเวลาของปีพุทธศักราชและแสดงผลลัพธ์ของปีที่เป็นปีอธิกสุรทินในช่วงเวลานั้น โดยมีหลักเกณฑ์ของปีอธิกสุรทินดังต่อไปนี้

การแบ่งปีตามหลักปฏิทินสากลแบ่งได้เป็นปีปกติมี 365 วัน และปีอธิกสุรทินมี 366 วัน โดยเพิ่มวันที่ 29 กุมภาพันธ์ ขึ้นมาอีกหนึ่งวัน สาเหตุที่กำหนดให้มีปีอธิกสุรทินนั้นเพราะโลกหมุนรอบดวงอาทิตย์หนึ่งรอบใช้เวลา 365.24218... วัน ดังนั้นจึงต้องมีการปรับให้จำนวนวันที่ปีไม่คลาดเคลื่อนกับวิถีการโคจร โดยปีที่เป็นปีอธิกสุรทินจะเป็นไปตามเงื่อนไขใดเงื่อนไขหนึ่งต่อไปนี้คือ

1. ปี ค.ศ. นั้นหารลงตัวด้วย 4 และหารไม่ลงตัวด้วย 100
2. ปี ค.ศ. นั้นหารลงตัวด้วย 400

จากกฎเกณฑ์นี้

- ปี พ.ศ. 2543 (ค.ศ. 2000) เป็นปีอธิกสุรทิน มี 366 วัน
- ปี พ.ศ. 2443 (ค.ศ. 1900) และ พ.ศ. 2643 (ค.ศ. 2100) เป็นปีปกติที่มี 365 วัน

ข้อมูลนำเข้า

- ปีพุทธศักราชเริ่มต้นและปีพุทธศักราชสุดท้าย

ข้อมูลส่งออก

- ปีอธิกสุรทินในช่วงเวลาตามข้อมูลนำเข้า
- หากไม่พบปีอธิกสุรทินให้แสดงเป็นคำว่า **None**

การประมวลผลข้อมูล

- กำหนดให้สร้างฟังก์ชันชื่อ **leaps** เพื่อใช้คำนวณหาปีอธิกสุรทินในช่วงเวลาตามข้อมูลนำเข้า และกำหนดให้คำนวณหาปีอธิกสุรทินโดยใช้ List comprehension เท่านั้น

หมายเหตุ

1. เครื่องหมาย ">" แสดงจุดเริ่มต้นของบรรทัด
2. ค่าที่ขีดเส้นใต้ แสดงข้อมูลที่ผู้ใช้ป้อน

ตัวอย่างการรันที่ 1

```
>Begin Year: 2500
>End Year: 2600
>Leap years from 2500 to 2600: 2503, 2507, 2511, 2515, 2519, 2523, 2527, 2531, 2535, 2539,
2543, 2547, 2551, 2555, 2559, 2563, 2567, 2571, 2575, 2579, 2583, 2587, 2591, 2595, 2599
```

ตัวอย่างการรันที่ 2

```
>Begin Year: 2500
>End Year: 2502
>Leap years from 2500 to 2502: None
```

17. [5 คะแนน] เขียนฟังก์ชัน `myStats` เพื่อคำนวณหาค่าพิสัย (Range) และค่าเฉลี่ย (Mean) ของข้อมูลที่กำหนดให้ โดยฟังก์ชันจะหาค่าพิสัย และค่าเฉลี่ยของข้อมูลแต่ละ tuple แล้วแสดงผลการคำนวณค่าพิสัยและค่าเฉลี่ยตามลำดับดังข้อมูลส่งออก

ข้อมูลนำเข้า

- ไม่มี

ข้อมูลส่งออก

- [(6, 5.0), (3, 6.5)]

ข้อมูลที่ใช้ในการคำนวณ

- `data = [(8, 2, 6, 4), (5, 6, 7, 8)]`

18. [5 คะแนน] เขียนฟังก์ชัน `encode` ซึ่งรับพารามิเตอร์เป็นสตริง `s` และส่งคืนค่าเป็นสตริง โดยฟังก์ชัน `encode` จะเปลี่ยนตัวอักษรแต่ละตัวในสตริง `s` เป็นรหัส ASCII 3 หลัก เช่น 'A' จะถูก เปลี่ยนเป็น '065' และ 'z' จะถูกเปลี่ยนเป็น '122' เป็นต้น

ตัวอย่างการเรียกใช้ฟังก์ชัน `encode`

Function call	Output
<code>print(encode('Cat'))</code>	067097116
<code>print(encode('Dog'))</code>	068111103
<code>print(encode('Bee'))</code>	066101101
<code>print(encode('Ant'))</code>	065110116

19. [5 คะแนน] เขียนฟังก์ชัน `isSorted` ซึ่งรับพารามิเตอร์เป็นลิสต์ `l` ของจำนวน และส่งคืนค่าเป็น `True` ถ้าค่าในลิสต์ `l` มีค่าแตกต่างกันทีละเรียงจากน้อยไปมาก มิฉะนั้นให้ส่งคืนค่าเป็น `False`

ในกรณีลิสต์ `l` มีสมาชิกเพียง 1 ตัวหรือเป็นลิสต์ว่างเปล่า ให้ส่งคืนค่าเป็น `True`

ตัวอย่างการเรียกใช้ฟังก์ชัน `isSorted`

Function call	Output
<code>print(isSorted([]))</code>	True
<code>print(isSorted([1]))</code>	True
<code>print(isSorted([1, 1]))</code>	False
<code>print(isSorted([1.1, 2.2]))</code>	True
<code>print(isSorted([0, -1, 5]))</code>	False
<code>print(isSorted([-5, -2, 0]))</code>	True

20. [10 คะแนน] เขียนโปรแกรมเพื่อฝึกการบวกเลข โดยมีเงื่อนไขดังต่อไปนี้

- ตัวเลขที่นำมาบวกกันจะเป็นจำนวนเต็มที่อยู่ระหว่าง 1 ถึง 10 (รวม 1 และ 10 ด้วย)
- โจทย์แต่ละข้อจะสุ่มจำนวนตัวเลขที่นำมาบวกกันอย่างน้อย 2 จำนวน แต่ไม่เกิน 5 จำนวน
- ถ้าผู้เล่นตอบถูกต้องจะได้ข้อละ 1 คะแนน
- ถ้าผู้เล่นตอบผิด โปรแกรมจะเฉลยคำตอบที่ถูกต้อง
- ถ้าผู้เล่นตอบผิดครบ 3 ข้อ โปรแกรมจะจบการทำงานและสรุปคะแนนที่ผู้เล่นได้

หมายเหตุ

1. เครื่องหมาย ">" แสดงจุดตั้งต้นของบรรทัด
2. ค่าที่ขีดเส้นใต้ แสดงข้อมูลที่ผู้ใช้ป้อน

ตัวอย่างการรัน

```
>7 + 1 + 4 = 12
>9 + 4 = 13
>2 + 6 + 10 + 9 + 1 = 28
>6 + 10 = 1
>Wrong! The answer was 16
>5 + 3 + 9 + 4 = 21
>8 + 5 = 13
>5 + 10 + 3 + 2 + 5 = 2
>Wrong! The answer was 25
>5 + 4 + 2 + 6 = 4
>Wrong! The answer was 17
>You earned 5 total points
```

ให้เขียนโปรแกรมโดยแบ่งเป็น 2 ฟังก์ชัน ได้แก่

- ฟังก์ชัน `question()` ซึ่งจะสร้างคำถาม 1 ข้อ และรับคำตอบจากผู้เล่น ถ้าตอบถูกต้องจะส่งคืนค่ากลับ (return) เป็น **True** แต่ถ้าตอบผิดจะเฉลยคำตอบและส่งคืนค่ากลับ (return) เป็น **False**
- ฟังก์ชัน `play()` ซึ่งจะเรียกฟังก์ชัน `question()` เรื่อย ๆ ไปจนกว่าผู้เล่นจะตอบผิดครบ 3 ข้อ และแสดงคะแนนที่ได้

สามารถใช้ฟังก์ชัน `randint()` จากไลบรารี `random` ซึ่งมีคำอธิบายการใช้งานดังนี้

`random.randint(a, b)` # ส่งคืนค่าเป็นจำนวนเต็ม N แบบสุ่มโดยที่ $a \leq N \leq b$

TABLE 1: Operator precedence and associativity

Level	Operators	Description	Associativity
1	(expressions...), [expressions...], {key:value...}, {expressions...}	binding or tuple display, list display, dictionary display, set display	left to right
2	x[index], x[index:index], x(arguments...), x.attribute	subscription, slicing, call, attribute reference	left to right
3	await x	await expression	left to right
4	**	exponentiation	right-to-left
5	+x, -x, ~x	positive, negative, bitwise NOT	left-to-right
6	*, @, /, //, %	multiplication, matrix multiplication, division, floor division, remainder	left-to-right
7	+x, -x	addition, subtraction	left to right
8	<<, >>	shifts	left to right
9	&	bitwise AND	left to right
10	^	bitwise XOR	left to right
11		bitwise OR	left to right
12	in, not in, is, is not, <, <=, >, >=, !=, ==	comparisons, including membership tests and identity tests	left to right
13	not x	boolean NOT	left to right
14	and	boolean AND	left to right
15	or	boolean OR	left to right
16	if - else	conditional expression	left to right
17	lambda	lambda expression	left to right

Operators are shown in decreasing order of precedence from level 1 to level 17.

TABLE 2: Standard ASCII code (in decimal)

Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	
13	CR	29	GS	45	-	61	=	77	M	93]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

SP means space.

ข้อสอบชุด B

กระดาษคำตอบ

การทดสอบความรู้ขั้นปริญญาตรี

ปลายภาคการศึกษาที่ 1 ปีการศึกษา 2561

วิชา วพ.101 การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น

วันศุกร์ที่ 14 ธันวาคม พ.ศ. 2561

เวลา 13:00 - 16:00 น.

ตอนที่:

☐ 810001 จ. 09:30 - 12:30 รศ. ดร. วีรชัย อโณทัยไพบูลย์

☐ 760001 อ. 09:30 - 12:30 อ. วชิรา พรหมสาขา ณ สกลนคร

ชื่อ-นามสกุล: _____ เลขทะเบียน: _____ เลขที่นั่งสอบ: _____

คะแนน

เติมคำ (15)	โปรแกรม (30)	รวม (45)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

ชื่อ-นามสกุล: _____ เลขทะเบียน: _____ เลขที่นั่งสอบ: _____

16. [5 คะแนน] โปรแกรม:

This image shows a single page of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

ชื่อ-นามสกุล:

เลขทะเบียน:

เลขที่นั่งสอบ:

17. [5 คะแนน] ฟังก์ชัน:

```
data = [(8,2,6,4),(5,6,7,8)]
print(myStats(data))
```

ชื่อ-นามสกุล: _____ เลขทะเบียน: _____ เลขที่นั่งสอบ: _____

18. [5 คะแนน] ฟังก์ชัน

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slightly aged or off-white appearance. The edges of the paper are visible, showing it's a full sheet. There is no handwriting or other markings on the paper.

ชื่อ-นามสกุล: _____ เลขทะเบียน: _____ เลขที่นั่งสอบ: _____

19. [5 คะแนน] ฟังก์ชัน:

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slightly aged or off-white appearance. The edges of the paper are visible, showing it's a full sheet. There is no handwriting or other markings on the paper.

ชื่อ-นามสกุล: _____ เลขทะเบียน: _____ เลขที่นั่งสอบ: _____

20. [10 คะแนน] โปรแกรม:

```
import random
```

```
def play():
```

```
    # Your code
```

```
def question():
```

```
    # Your code
```

```
play()
```