

แบบฝึกหัดชุดที่ 4

1. จงเขียนโปรแกรมคำนวณรายได้จากเงินฝากพร้อมดอกเบี้ยเมื่อเวลาผ่านไป 1 ปี, 2 ปี, 3 ปี, และ 4 ปีที่คำนวณเงินได้แบบดอกเบี้ยทบต้น โดยมีข้อมูลเข้า คือ อัตราดอกเบี้ย และแสดงผลลัพธ์จากการคำนวณ โดยพิมพ์ผลลัพธ์แบบชิดขวาที่แสดงเลขทศนิยม 2 ตำแหน่ง เมื่อ

$$\text{รายได้จากเงินฝาก} = \text{เงินต้น}(M) \times (1 - \text{อัตราดอกเบี้ย}(R))^p = M \times (1+R)^y$$

ให้แสดงรายได้สำหรับเงินต้น 10,000 15,000 20,000 25,000 30,000 35,000 40,000

ผลลัพธ์

Enter interest rate: 5

year	1	2	3	4
10000.00	10500.00	11025.00	11576.25	12155.06
15000.00	15750.00	16537.50	17364.38	18232.59
20000.00	21000.00	22050.00	23152.50	24310.13
25000.00	26250.00	27562.50	28940.63	30387.66
30000.00	31500.00	33075.00	34728.75	36465.19
35000.00	36750.00	38587.50	40516.88	42542.72
40000.00	42000.00	44100.00	46305.00	48620.25

```
แบบฝึกหัดที่ 4 > 4.1.py
1  r=float(input("Enter interest rate: "))
2  m=[10000,15000,20000,25000,30000,35000,40000]
3  y=[1,2,3,4]
4  total=[]
5  for i in range(len(y)):
6      for n in range(len(m)):
7          sum = float(m[n])*((1+r/100)**int(y[i]))
8          total.append("%.2f"%sum)
9  print(f" year      1      2      3      4")
10 for i in range(7):
11     print(f"{m[i]}.00|{total[i]}|{total[i+7]}|{total[i+14]}|{total[i+21]}|")
12
```

```
Enter interest rate: 5
year      1      2      3      4
10000.00|10500.00|11025.00|11576.25|12155.06|
15000.00|15750.00|16537.50|17364.38|18232.59|
20000.00|21000.00|22050.00|23152.50|24310.13|
25000.00|26250.00|27562.50|28940.63|30387.66|
30000.00|31500.00|33075.00|34728.75|36465.19|
35000.00|36750.00|38587.50|40516.88|42542.72|
40000.00|42000.00|44100.00|46305.00|48620.25|
```

2. ให้รับเวลาเข้าและออกของรถคันหนึ่ง (เปิดบริการตั้งแต่ 7:00 - 23:00) จากนั้นคำนวณค่าที่จอดรถที่ต้องจ่าย โดยหลักเกณฑ์การคำนวณมีดังนี้ (สมมติว่าไม่มีการจอดข้ามวัน)

- จอดรถไม่เกิน 15 นาที ไม่คิดค่าบริการ
- จอดรถเกิน 15 นาที แต่ไม่เกิน 3 ชั่วโมง คิดค่าบริการชั่วโมงละ 10 บาท เศษของชั่วโมงคิดเป็นหนึ่งชั่วโมง
- จอดรถตั้งแต่ 4 ชั่วโมง ถึง 6 ชั่วโมง คิดค่าบริการชั่วโมงที่ 4-6 ชั่วโมงละ 20 บาท เศษของชั่วโมงคิดเป็นหนึ่งชั่วโมง
- จอดรถเกิน 6 ชั่วโมงขึ้นไป เหม่าจ่ายวันละ 200 บาท

ข้อมูลนำเข้า

มี 4 บรรทัด แต่ละบรรทัดมีจำนวนเต็มหนึ่งจำนวน

โดยบรรทัดที่ 1-2 เป็นชั่วโมงและนาทีของเวลาเข้า และบรรทัดที่ 3-4 เป็นชั่วโมงและนาทีของเวลาออก

ข้อมูลส่งออก

มีบรรทัดเดียว เป็นค่าที่จอดรถที่ต้องจ่าย ให้แสดงผลลัพธ์เป็นจำนวนเต็ม

ตัวอย่าง

7	7	7	7
0	0	30	30
7	7	10	13
15	16	31	31
* => 0	* => 10	* => 50	* => 200

```

1 in_hour = int(input("Enter hour_in: "))
2 in_min = int(input("Enter minute_in: "))
3 out_hour = int(input("Enter hour_out: "))
4 out_min = int(input("Enter minute_out: "))
5 hour = out_hour-in_hour
6 min = out_min-in_min
7 if in_hour>=7 and out_hour<=23 and out_min<60 and in_min<60:
8     if hour>=1 and out_min-in_min<=0:
9         min = 60-(in_min-out_min)
10        hour=hour-1
11        if hour<0:
12            hour=0
13    elif min<0 and out_min!=0:
14        min = in_min+out_min
15        if min>59:
16            min=min-58
17            hour=hour-1
18            if hour <0:
19                hour = 0
20    elif min<0 and out_min==0:
21        min = 60-in_min
22        hour=hour-1
23        if hour <0:
24            hour = 0
25 else:
26     print("เปิดบริการเวลา 7:00น. - 23:00น. ขออภัยในความไม่สะดวก กรุณาใส่เวลาอีกครั้ง")
27     print("เปลี่ยนบริการเวลา 7:00น. - 23:00น. ขออภัยในความไม่สะดวก กรุณาใส่เวลาอีกครั้ง")
28 if min<15 and hour==0 and in_hour>=7 and out_hour<=23 and out_min<60 and in_min<60:
29     print("ไม่คิดค่าบริการ")
30 elif min>0 and in_hour>=7 and out_hour<=23 and out_min<60 and in_min<60:
31     hour=hour+1
32     if hour<=3:
33         money = 10*hour
34     elif hour>=4 and hour<=6:
35         money = 20*hour
36     elif hour>6:
37         money = 200
38     else:
39         print("Error")
40     print(f"ค่าจอดรถที่ต้องจ่าย {money}บาท")
41 else:
42     print("Error")
43

```

```

Enter hour_in: 7
Enter minute_in: 0
Enter hour_out: 7
Enter minute_out: 15
ไม่คิดค่าบริการ
PS C:\Users\KMITL64015172\OneDrive - KMITL\Documents> python 72/AppData/Local/Programs/Python/Python39/python.exe
Documents/py/แบบฝึกหัดที่ 4/4.2.py"
Enter hour_in: 7
Enter minute_in: 30
Enter hour_out: 13
Enter minute_out: 31
ค่าจอดรถที่ต้องจ่าย 200บาท

```

3. 2520 คือ ตัวเลขที่น้อยที่สุด ที่สามารถหารด้วยตัวเลขทุกตัวตั้งแต่ 1-10 จงหาจำนวนเต็มบวกที่น้อยที่สุดที่หารด้วยตัวเลขทุกตัวตั้งแต่ 1-20

```
แบบฝึกหัดที่ 4 > 4.3.py > ...
1 list=[2,3,5,7,11,13,17,19]
2 num=1
3 for i in range(8):
4     numlist = list[i]
5     n=1
6     while True:
7         if (numlist**n)>20:
8             num*= numlist**(n-1)
9             break
10            n=n+1
11 print(num)
```

```
Documents/py/แบบฝึกหัดที่ 4/4.3.py"
232792560
```

4. prime factors คือ ตัวเลขจำนวนเฉพาะที่คูณกันแล้วได้เท่ากับจำนวนที่กำหนด เช่น prime factors ของ 13195 คือ 5, 7, 13 และ 29 ให้เขียนโปรแกรมหา prime factor ของ 600851475143

```
แบบฝึกหัดที่ 4 > 4.4.py > ...
1 list=[]
2 num=600851475143
3 n=2
4 while num!=1:
5     if(num%n)==0:
6         list.append(n)
7         num = num/n
8     n = n+1
9 print(*list,sep=", ")
```

```
Documents/py/แบบฝึกหัดที่ 4/4.4.py"
71, 839, 1471, 6857
```

5. จำนวนเฉพาะ (Prime Number) คือตัวเลขที่มีแต่ 1 กับตัวมันเองที่หารลงตัว โดยจำนวนเฉพาะ 6 ตัวแรก คือ 2, 3, 5, 7, 11, 13 โดยจำนวนเฉพาะตัวที่ 6 คือ 13 จงหาจำนวนเฉพาะตัวที่ 1001

```
แบบฝึกหัดที่ 4 > 4.5.py > ...
1 count = 1
2 i = 3
3 while count != 1001:
4     for k in range(2,i):
5         if i%k == 0:
6             break
7     else:
8         count += 1
9     i += 2
10 print(f"Prime Number 1001 : {i}")
```

```
Documents/py/แบบฝึกหัดที่ 4/4.5.py"
Prime Number 1001 : 7929
```

6. sum of the squares ของ 1-10 คือ

$$1^2 + 2^2 + \dots + 10^2 = 385$$

ส่วน square of the sum 1-10 คือ

$$(1 + 2 + \dots + 10)^2 = 55^2 = 3025$$

ผลต่างระหว่าง square of the sum กับ sum of the squares = $3025 - 385 = 2640$ ให้หาผลต่างของ square of the sum กับ sum of the squares ของ 1-100

```
แบบฝึกหัดที่ 4 > 4.6.py > ...
1 num=int(input("Enter value of num: "))
2 sum_of_squares = 0
3 for n in range(1,num+1):
4     sum_of_squares = sum_of_squares+(n*n)
5 print(f"Sum of squares = {sum_of_squares}")
6
7 sum = 0
8 for i in range(1,num+1):
9     sum = sum+i
10    square_of_the_sum = sum*sum
11 print(f"Square of the sum = {square_of_the_sum}")
12 net = square_of_the_sum -sum_of_squares
13 print(f"ผลต่างของ square of the sum กับ sum of the squares : {net}")
```

```
Documents/py/แบบฝึกหัดที่ 4/4.6.py"
Enter value of num: 100
Sum of squares = 338350
Square of the sum = 25502500
ผลต่างของ square of the sum กับ sum of the squares : 25164150
```

7. จากตัวเลขต่อไปนี้ ตัวเลขติดกัน 4 ตัวที่เมื่อนำมาคูณกันแล้วมีค่ามากที่สุดคือ $9 \times 9 \times 8 \times 9 = 5832$

73167176531330624919225119674426574742355349194934
96983520312774506326239578318016984801869478851843
85861560789112949495459501737958331952853208805511
12540698747158523863050715693290963295227443043557
66896648950445244523161731856403098711121722383113
62229893423380308135336276614282806444486645238749
30358907296290491560440772390713810515859307960866
70172427121883998797908792274921901699720888093776
65727333001053367881220235421809751254540594752243
52584907711670556013604839586446706324415722155397
53697817977846174064955149290862569321978468622482
83972241375657056057490261407972968652414535100474
82166370484403199890008895243450658541227588666881
16427171479924442928230863465674813919123162824586
17866458359124566529476545682848912883142607690042
24219022671055626321111109370544217506941658960408
07198403850962455444362981230987879927244284909188
84580156166097919133875499200524063689912560717606
05886116467109405077541002256983155200055935729725
71636269561882670428252483600823257530420752963450

จงหาเลขติดกัน 8 ตัวที่เมื่อนำมาคูณกันแล้วมีค่ามากที่สุด และเป็นเลขอะไร ผลคูณเท่ากับเท่าไร

```

1 num = "731671765313306249192251196744265747423553491949349698352031277450632623957831801698480"
2 max = 0
3 list=[0,0,0,0,0,0,0,0]
4 for i in range(len(num)-8):
5     add = 1
6     no = num[i:i+8]
7     for i in range(8):
8         add*=int(no[i])
9     if add>max:
10        max=add
11        for i in range(8):
12            list[i]=no[i]
13 for n,i in enumerate(list):
14     if n==len(list)-1:
15         print(i)
16     else:
17         print(i,end="x")
18 print(f"= {max}",end="")

```

8x8x3x9x9x8x7x9
= 7838208

8. ให้เขียนโปรแกรมรับข้อมูล 1 บรรทัด ประกอบด้วยตัวเลข 1 หลัก จำนวนไม่เกิน 10 ตัว คั่นด้วยช่องว่าง จากนั้นให้นำตัวเลขที่รับเข้ามาเรียงกัน และหาลำดับการเรียงที่ทำให้มีค่าน้อยที่สุด โดยต้องไม่ขึ้นต้นด้วย 0

Input : 9 4 6 2 คำตอบ 2469, Input : 3 0 8 1 3 3 คำตอบ : 103338

```

1 Num = [int(e) for e in input("Enter 10 Number: ").split()]
2 list = []
3 list_num = []
4 for i in range(0,len(Num)):
5     if Num[i] == 0:
6         list.append(Num[i])
7     for i in range(0,len(Num)):
8         if Num[i] != 0:
9             list_num.append(Num[i])
10 Boolean = True
11 while Boolean:
12     Boolean1 = False
13     for i in range(0,len(list_num)):
14         if i + 1 > len(list_num)-1:
15             break
16         elif list_num[i] > list_num[i+1]:
17             list_num[i],list_num[i+1] = list_num[i+1],list_num[i]
18             Boolean1 = True
19     Boolean = Boolean1
20 for i in list:
21     list_num.insert(1,0)
22 print("คำตอบ: ",*list_num,sep="")

```

Documents/py/แบบฝึกหัดที่ 4/4.8.py"

Enter 10 Number: 3 0 8 1 3 3

คำตอบ: 103338

9. ตัวเลข palindrome คือตัวเลขที่อ่านได้ทั้ง 2 ทาง แล้วมีค่าเท่ากัน เช่น 9009 โดย 9009 คือ palindrome ที่เกิดจากการคูณของตัวเลข 2 หลักที่มากที่สุด คือ 91x99 จงหา palindrome ที่มากที่สุดของตัวเลข 3 หลัก

```

1 n = 0
2 for a in range(999, 100, -1):
3     for b in range(a, 100, -1):
4         x = a * b
5         if x > n:
6             s = str(a * b)
7             if s == s[::-1]:
8                 n = a * b
9 print(f"palindrome ที่มากที่สุดของตัวเลข 3 หลัก คือ {n}")

```

Documents/py/แบบฝึกหัดที่ 4/4.9.py"

palindrome ที่มากที่สุดของตัวเลข 3 หลัก คือ 906609

