

#### แบบฝึกหัดชุดที่ 4

1. จงเขียนโปรแกรมคำนวณรายได้จากเงินฝากพร้อมดอกเบี้ยเมื่อเวลาผ่านไป 1 ปี, 2 ปี, 3 ปี, และ 4 ปีที่  
คำนวณเงินได้แบบดอกเบี้ยทบต้น โดยมีข้อมูลเข้า คือ อัตราดอกเบี้ย และแสดงผลจากการ  
การคำนวณ โดยพิมพ์ผลลัพธ์แบบชิดขวาที่แสดงเลขทศนิยม 2 ตำแหน่ง เมื่อ

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

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 |

```
ex2.py > ...
1  r = int(input("Enter Number : "))
2  m = [10000, 15000, 20000, 25000, 30000, 35000, 40000]
3  y = [1, 2, 3, 4]
4  final = []
5  for a in range(len(y)) :
6      for b in range(len(m)) :
7          sum = int(m[b])*((1+r/100)**int(y[a]))
8          final.append('%.2f' %sum)
9
10 print(f"year    1        2        3        4")
11 for i in range(7):
12     print(f"{m[i]}|{final[i]}|{final[i+7]}|{final[i+14]}|{final[i+21]}|")
```

```
Enter Number : 5
year    1        2        3        4
10000|10500.00|11025.00|11576.25|12155.06|
15000|15750.00|16537.50|17364.38|18232.59|
20000|21000.00|22050.00|23152.50|24310.13|
25000|26250.00|27562.50|28940.63|30387.66|
30000|31500.00|33075.00|34728.75|36465.19|
35000|36750.00|38587.50|40516.88|42542.72|
40000|42000.00|44100.00|46305.00|48620.25|
PS D:\Python>
```

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 |

```

ex2.py > ...
1 h1 = int(input("First Hour :"))
2 m1 = int(input("First Minute :"))
3 h2 = int(input("Second Hour :"))
4 m2 = int(input("Second Minute :"))
5 hour = h2-h1
6 minute = m2-m1
7 point = 0
8
9 if not(hour == 0 and minute <= 15) :
10     if hour == 0 and minute > 15 or hour == 1 and minute == 0 :
11         hour = 1
12         point = 10*hour
13     elif hour == 1 and minute > 0 or hour == 2 and minute == 0 :
14         hour = 2
15         point = 10*hour
16     elif hour == 2 and minute > 0 or hour == 3 and minute == 0 :
17         hour = 3
18         point = 10*hour
19     elif hour == 3 and minute > 0 or hour == 4 and minute == 0 :
20         hour = 4
21         hour = hour-3
22         point = 30+(20*hour)
23     elif hour == 4 and minute > 0 or hour == 5 and minute == 0 :
24         hour = 5

```

```

23 elif hour == 4 and minute > 0 or hour == 5 and minute == 0 :
24     hour = 5
25     hour = hour-3
26     point = 30+(20*hour)
27 elif hour == 5 and minute > 0 or hour == 6 and minute == 0 :
28     hour = 6
29     point = 200
30 elif hour >= 6 :
31     point = 200
32 print(f"ราคา = {point} บาท")
33 else :
34     print("Free")

```

```

First Hour :7
First Minute :30
Second Hour :10
Second Minute :31
ราคา = 50 บาท
PS D:\Python>

```

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

```
ex2.py > ...
1 prime_factors = [2, 3, 5, 7, 11, 13, 17, 19]
2 sum = 1
3
4 for i in range(len(prime_factors)) :
5     n = int(prime_factors[i])
6     a = 1
7     while True :
8         if n**a > 20 :
9             sum *= n**(a-1)
10            break
11        a += 1
12 print("Sum : ",int(sum))
```

```
Sum : 232792560
PS D:\Python> 
```

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

```
ex2.py > ...
1 prime_factors = int(input("Enter Number : "))
2 prime_factors_list = []
3
4 i = 2
5 while True :
6     if prime_factors % i == 0 :
7         prime_factors = prime_factors / i
8         prime_factors_list.append(i)
9     elif(prime_factors < i) :
10        break
11    else :
12        i += 1
13 print(f"Prime Factors = {prime_factors_list[-1]}")
14
```

```
Enter Number : 600851475143
Prime Factors = 6857
PS D:\Python> 
```

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

```
ex2.py > ...
1 prime_factors_list = []
2
3 i = 2
4 while len(prime_factors_list) < 1001 :
5     j = 2
6     while j<=i//j:
7         if i%j==0:
8             break
9         j+=1
10    if j > i//j:
11        prime_factors_list.append(i)
12    i += 1
13 print(f"Prime Factors = {prime_factors_list[-1]}")
```

```
Prime Factors = 7927
PS D:\Python> 
```

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

```
ex2.py > ...
1  num = int(input("Number = "))
2  sum_of_the_squares = 0
3  square_of_the_sum1 = 0
4
5  for i in range(1,num+1) :
6      sum_of_the_squares += i**2
7      square_of_the_sum1 += i
8      square_of_the_sum = square_of_the_sum1**2
9      sum = square_of_the_sum - sum_of_the_squares
10
11 print(f"{{square_of_the_sum}} - {{sum_of_the_squares}} = {{sum}}")
```

```
Number = 100
25502500 - 338350 = 25164150
PS D:\Python> |
```

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

```

ex2.py > ...
1 data_1 = ''73167176531330624919225119674426574742355349194934
2 96983520312774506326239578318016984801869478851843
3 85861560789112949495459501737958331952853208805511
4 12540698747158523863050715693290963295227443043557
5 66896648950445244523161731856403098711121722383113
6 6222989342338030813533627661428280644486645238749
7 30358907296290491560440772390713810515859307960866
8 70172427121883998797908792274921901699720888093776
9 65727333001053367881220235421809751254540594752243
10 52584907711670556013604839586446706324415722155397
11 53697817977846174064955149290862569321978468622482
12 83972241375657056057490261407972968652414535100474
13 8216637048440319989008895243450658541227588666881
14 16427171479924442928230863465674813919123162824586
15 17866458359124566529476545682848912883142607690042
16 2421902267105562632111109370544217506941658960408
17 07198403850962455444362981230987879927244284909188
18 84580156166097919133875499200524063689912560717606
19 05886116467109405077541002256983155200055935729725
20 71636269561882670428252483600823257530420752963450'''

```

```

21
22 data = data_1.replace("\n","")
23 max = 1
24 list_max = []
25 for i in range(0,len(data)) :
26     word = data[i:i+8]
27     before_max = 1
28     for j in range(0,len(word)) :
29         before_max *= int(word[j])
30
31     if (before_max > max) :
32         list_max.clear()
33         list_max.append(word)
34         max = before_max
35
36
37 print(f"{list_max} = {max}")

```

```

['88399879'] = 7838208
PS D:\Python>

```

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

```

ex2.py > ...
1 numinput = [int(x) for x in input("Number : ").split()]
2 numbers = []
3
4 for i in range(0,len(numinput)) :
5     if(numinput[i] != 0) :
6         numbers.append(numinput[i])
7
8 zero_number = []
9
10 for j in range(0,len(numinput)) :
11     if(numinput[j] == 0) :
12         zero_number.append(numinput[j])
13
14 check = True
15 while check :
16     check_2 = False
17     for i in range(0,len(numbers)) :
18         if(i+1 > len(numbers)-1) :
19             break
20         elif(numbers[i]>numbers[i+1]) :
21             numbers[i],numbers[i+1] = numbers[i+1],numbers[i]
22             check_2 = True
23
24     check = check_2
25
26 for j in zero_number :
27     numbers.insert(1,zero_number[j])
28
29 print(numbers)
30

```

```

Number : 3 0 8 1 3 3
[1, 0, 3, 3, 3, 8]
PS D:\Python>

```

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

```

ex2.py > ...
1 palindrome = []
2 sum1 = 0
3 for i in range(100,1000) :
4     for j in range(100,1000) :
5         sum1 = i*j
6         sum = str(sum1)
7         rev = sum[::-1]
8         if sum == rev :
9             palindrome.append(int(sum))
10 print(f"Palindrom = {max(palindrome)}")

```

```

Palindrom = 906609
PS D:\Python>

```