Assignment #4

64015030 ชินพัฒน์ ลิ้มประธาน

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

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

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

```python
rate = int(input("Enter interest rate : ")) / 100

money_list = [10000, 15000, 20000, 25000, 30000, 35000, 40000]

print(f"{'year':10}{'1':^12}{'2':^10}{'3':^12}{'4':^10}")

for money in money_list:
    interest = (1+rate)
    print(f"{money:.2f} | {money * (interest**1):.2f} | {money * (interest**2):.2f} | {money * (interest**3):.2f} |
    {money * (interest**4):.2f}")
```

```
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 ชั่วโมง ละ 20 บาท เศษของชั่วโมงคิดเป็นหนึ่งชั่วโมง

- จอดรถเกิน 6 ชั่วโมงขึ้นไป เหมาจ่ายวันละ 200 บาท

ข้อมูลนำเข้า

มี 4 บรรทัด แต่ละบรรทัดมีจำนวนเต็มหนึ่งจำนวน

โดยบรรทัดที่ 1-2 เป็นชั่วโมงและนาทีของเวลาเข้า และบรรทัดที่ 3-4 เป็นชั่วโมงและนาทีของเวลาออก

ข้อมูลส่งออก

มีบรรทัดเดียว เป็นค่าที่จอดรถที่ต้องจ่าย ให้แสดงผลลัพธ์เป็นจำนวนเต็ม

```python
hour_enter = int(input(""))
minute_enter = int(input(""))
hour_exit = int(input(""))
minute_exit = int(input(""))

if (hour_enter < 7 or hour_exit ≥ 23) or (minute_enter < 0 or minute_exit > 59):
    print("invalid hour please park between (7:00 - 23:00):")
else:
    if minute_exit == minute_enter:
        total_hours = hour_exit - hour_enter
        total_minutes = 0
    elif minute_exit > minute_enter:
        total_hours = hour_exit - hour_enter
        total_minutes = minute_exit - minute_enter
    else:
        total_hours = (hour_exit - hour_enter) - 1
        total_minutes = 60 - (minute_enter - minute_exit)

    parking_cost = 0

    if total_minutes ≤ 15 and total_hours ≤ 0:
        parking_cost = 0
    elif total_hours ≥ 6 and total_minutes > 0:
        parking_cost = 200
    elif total_hours ≤ 3:
        if total_minutes > 0 and total_hours == 3:
            total_hours += 2
        elif total_minutes > 0:
            total_hours += 1
        parking_cost = total_hours * 10
    elif 4 ≤ total_hours ≤ 6:
        if total_minutes > 0:
            total_hours += 1
        parking_cost = total_hours * 20

    print(f"=> {parking_cost}")
```

```
7
0
10
31
=> 50
```

```
7
0
7
15
=> 0
```

```
7
30
13
31
=> 200
```

```
7
0
7
16
=> 10
```

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

```python
number = 1
is_twenty = False
while not is_twenty:
    number += 1
    for divisor in range(1, 21):
        if number % divisor == 0:
            is_twenty = True
        else:
            is_twenty = False
            break
print(number)
```

```
232792560
```

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

```python
number = 600851475143
prime_factors = []
sum_of_prime = 1
for divisor in range(2, number+1):
    if number % divisor == 0:
        prime_factors.append(divisor)
        sum_of_prime *= divisor

    if sum_of_prime == number:
        break

print(prime_factors[-1])
```

```
6857
```

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

```python
prime_position = 1
prime_value = 0
number = 2

while True:
    divided_count = 0
    if prime_position == 1002:
        break
    for divisor in range(1, number+1):
        if number % divisor == 0:
            divided_count += 1
    if divided_count == 2:
        prime_value = number
        prime_position += 1
    number += 1

print(f"position  => {prime_position-1} : value => {prime_value}")
```

```
position  => 1001 : value => 7927
```

6. sum of the squares ของ 1-10 คือ

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

ส่วน square of the sum 1-10 คือ

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

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

```python
sum_of_squares = 0
square_of_sum = 0

for number in range(1, 101):
    sum_of_squares += (number**2)
    square_of_sum += number

result = ((square_of_sum)**2) - sum_of_squares
print(result)
```

```
25164150
```

7. จากตัวเลขต่อไปนี้ ตัวเลขติดกัน 4 ตัวที่เมื่อนำมาคูณกันแล้วมีค่ามากที่สุดคือ 9 × 9 × 8 × 9 = 5832

73167176531330624919225119674426574742355349194934969835203127745063262395783180169848018694788518438586156078911294949545950173795833195285320880551112540698747158523863050715693290963295227443043557668966489504452445231617318564030987111217223831136222989342338030813533627661428280644448664523874930358907296290491560440772390713810515859307960866701724271218839987979087922749219016997208880937766572733300105336788122023542180975125454059475224352584907711670556013604839586446706324415722155397539783797784617406495514929086256932197846862248283972241375657056057490261407972968652414535100474821663704844031998900088952434506585412275886668811642717147992444292823086346567481391912316282458617866458359124566529476545682848912883142607690042421902267105562632111110937054421750694165896040807198403850962455444362981230987879927244284909188845801561660979191338754992005240636899125607176060588611646710940507754100225698315520005593572972571636269561882670428252483600823257530420752963450

จงหาเลขติดกัน 8 ตัวที่เมื่อนำมาคูณกันแล้วมีค่ามากที่สุด และเป็นเลขอะไร ผลคูณเท่ากับเท่าไร

```python
number_string = 
"731671765313306249192251196744265747423553491949349698352031277450632623957831801698480186947885184385861560789112949495459501737958331952853208805511125406987471585238630507156932909632952274430435576689664895044524452316173185640309871112172238311362229893423380308135336276614282806444486645238749303589072962904915604407723907138105158593079608667017242712188399879790879227492190169972088809377665727333001053367881220235421809751254540594752243525849077116705560136048395864467063244157221553975369781797784617406495514929086256932197846862248283972241375657056057490261407972968652414535100474821663704844031998900088952434506585412275886668811642717147992444292823086346567481391912316282458617866458359124566529476545682848912883142607690042421902267105562632111110937054421750694165896040807198403850962455444362981230987879927244284909188845801561660979191338754992005240636899125607176060588611646710940507754100225698315520005593572972571636269561882670428252483600823257530420752963450"

times_number = ''
max_times_values = 0
for number_i in range(len(number_string)):
    if number_i+7 >= len(number_string):
        break
    number = int(number_string[number_i]) * int(number_string[number_i+1]) * int(number_string[number_i+2]) * int(number_string[number_i+3]) * int(number_string[number_i+4]) * int(number_string[number_i+5]) * int(number_string[number_i+6]) * int(number_string[number_i+7])
    if number > max_times_values:
        times_number = number_string[number_i] + number_string[number_i+1] + number_string[number_i+2] + number_string[number_i+3] + number_string[number_i+4] + number_string[number_i+5] + number_string[number_i+6] + number_string[number_i+7]
        max_times_values = number

print(f"{' x '.join(times_number)} = {max_times_values}")
```

```
8 x 8 x 3 x 9 x 9 x 8 x 7 x 9 = 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

```python
number_list = []
number_list = [int(e) for e in input("").split()]
number_not_zero = []
if len(number_list) ≤ 10 and number_list[0] ≠ 0:
    ## select only number not zero
    for number in number_list:
        if number ≠ 0:
            number_not_zero.append(number)

    ## sort number
    for num_i in number_not_zero:
        for num_j in range(len(number_not_zero)):
            if num_j+1 ≥ len(number_not_zero):
                break
            if number_not_zero[num_j] > number_not_zero[num_j+1]:
                number_not_zero[num_j], number_not_zero[num_j+1] = number_not_zero[num_j+1], number_not_zero[num_j]

    ## insert value into list that is not zero
    arr_index = 0
    for num_i in range(len(number_list)):
        if number_list[num_i] ≠ 0:
            number_list[num_i] = number_not_zero[arr_index]
            arr_index += 1
    print(number_list)
else:
    print("Please enter number lower than 10")
```

```
3 0 8 1 3 3
[1, 0, 3, 3, 3, 8]
```

```
1 3 0 0 4 2 8 6 0
[1, 2, 0, 0, 3, 4, 6, 8, 0]
```

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

```python
palindrome = 0
max_palindrome = 0
main = 0
times = 0
for i in range(100, 1000):
    for j in range(100, 1000):
        number = str(i*j)
        number_reverse = number[::-1]
        if number_reverse == number:
            palindrome = int(number)
            if palindrome > max_palindrome:
                main = i
                times = j
                max_palindrome = palindrome
print(f"{main} x {times} = {max_palindrome}")
```

```
913 x 993 = 906609
```