금공프3 Quiz 3

MFE 20249433 최재필

Q1

1. Calculate the following by using 'while' loop.

$$\sum_{\substack{i \geq 1, \\ i^4 < 1000}}^{10} i^4$$

In [ ]:
```
i = 1
summation = 0

while i<=10 and i**4 < 1000:
    summation += i**4
    print(f'{i}: i^4={i**4} / Summation={summation}')
    i += 1
```

```
1: i^4=1 / Summation=1
2: i^4=16 / Summation=17
3: i^4=81 / Summation=98
4: i^4=256 / Summation=354
5: i^4=625 / Summation=979
```

Q2

2. Using the two list objects, x=[1, 2, 3, 4] & y = [5, 6, 7, 8], answer the following.

   (1) Write a Python program to create a list z which consists of the Cartesian product of x and y : [[1, 5], [1, 6],... ,[1, 8], [2, 5],...,[4, 5],...,[4, 8]]). Use list comprehension.

   (2) Modify the code in (1) so that only the elements with a sum of two numbers greater than or equal to 8 are left in z. Use list comprehension.

In [ ]:
```
x = [1, 2, 3, 4]
y = [5, 6, 7, 8]
```

(1)

In [ ]:
```
z = [[a, b] for a in x for b in y]
z
```

Out[ ]:
```
[[1, 5],
 [1, 6],
 [1, 7],
 [1, 8],
 [2, 5],
 [2, 6],
 [2, 7],
 [2, 8],
 [3, 5],
 [3, 6],
 [3, 7],
 [3, 8],
 [4, 5],
 [4, 6],
 [4, 7],
 [4, 8]]
```

(2)

In [ ]:
```
z = [[a, b] for a in x for b in y if (a+b) >= 8]
z
```

Out[ ]:
```
[[1, 7],
 [1, 8],
 [2, 6],
 [2, 7],
 [2, 8],
 [3, 5],
 [3, 6],
 [3, 7],
 [3, 8],
 [4, 5],
 [4, 6],
 [4, 7],
 [4, 8]]
```

Q3

Grading Score

In [ ]:
```
def grading(score):
    if not (0 <= score <= 100):
        return 'Score must be a number between 0 and 100!!'

    if 90 < score:
        grade = 'A'
    elif 80 < score:
        grade = 'B'
    elif 70 < score:
        grade = 'C'
    elif 60 < score:
        grade = 'D'
    else:
        grade = 'F'
```

```
    return f'Grade is {grade}!'
```

In [ ]:
```
grading(score=75)
```

Out[ ]:
```
'Grade is C!'
```

In [ ]:
```
grading(-5)
```

Out[ ]:
```
'Score must be a number between 0 and 100!!'
```

Q4

Explain why the error occrus

(1)

In [ ]:
```
def infoprint(name, age, gender):
    print(name, 'is', age, 'years old', gender, '.')
```

In [ ]:
```
infoprint(name='Kim', 'male')
```

```
  Cell In[20], line 1
    infoprint(name='Kim', 'male')
                                 ^
SyntaxError: positional argument follows keyword argument
```

Error occured because positional arguments should always come before keyword arguments

In [ ]:
```
# To fix this, correct the order and add a missing argument
infoprint('Kim', 13, gender='male')
```

```
Kim is 13 years old male .
```

(2)

In [ ]:
```
infoprint('Kim', gender='male')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[21], line 1
----> 1 infoprint('Kim', gender='male')

TypeError: infoprint() missing 1 required positional argument: 'age'
```

The function requires all three arguments: `name` , `age` , and `gender` inthe correct order.

In [ ]:
```
# To fix this, add a missing argument
infoprint('Kim', 13, gender='male')
```

```
Kim is 13 years old male .
```

(3)

In [ ]:
```
fac = 1

def myfactorial(n):
    for i in range(n):
        fac *= i + 1

    return fac
```

In [ ]:
```
myfactorial(n=5)
```

```
---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
Cell In[26], line 1
----> 1 myfactorial(n=5)

Cell In[25], line 5, in myfactorial(n)
      3 def myfactorial(n):
      4     for i in range(n):
----> 5         fac *= i + 1
      7     return fac

UnboundLocalError: cannot access local variable 'fac' where it is not associated
with a value
```

A variable `fac` was not declared inside a function, `myfactorial()`.

Either `global` should be used or `fac` should be declared inside the function.

In [ ]:
```
# To fix this (1)

fac = 1

def myfactorial(n):
    global fac
    for i in range(n):
        fac *= i + 1

    return fac
```

In [ ]:
```
myfactorial(n=5)
```

Out[ ]:
```
120
```

In [ ]:
```
# To fix this (2)

def myfactorial(n):
    fac = 1
    for i in range(n):
        fac *= i + 1

    return fac
```

In [ ]:
```
myfactorial(n=5)
```

```
Out[ ]: 120
```