코딩하기 좋은 날

For 양민 in 파이썬:
Return 전문가 (feat. 잠자기 좋은 프로그램)

"수학은 이해하는 것이 아니다. 그저 익숙해지는 것일 뿐이다." - John von Neumann

TEST _ 초급 수준 감 잡기

1. 다음 중 알맞은 설명은?
   ①파이썬에서 변수명은 camelCASE로 SomeVariablename처럼 짓는다
   ②파이썬에서 변수명은 camelCASE로 SomeFuntionname처럼 짓는다
   ③파이썬에 상수명은 대문자로 INTREST_RATE처럼 써주어야 한다.
   ④파이썬에서 한 프로그램내에서 상수명을 정의한 후 바꾸면 오류가 난다.
   ⑤파이썬에서 코멘트를 달아야할 경우 ,//를 쓰고 설명을 달면 된다.

2. 출력값은?
   ① 9
   ② NONE
   ③ NULL
   ④ 9 ₩n NONE
   ⑤ 0 ₩n NULL

```
def print_square(x):
    print(x * x)

print(print_square(3))
```

3. 출력값은?
   ① [1,2,3,4]
   ②[1,2,3,4,5,]
   ③[5,6,7]
   ④[4,5,6,7]
   ⑤ ERROR 발생

```
numbers = [1, 2, 3, 4, 5, 6, 7]
print(numbers[0:-3])
```

1. (4) 2. (4) 3. (1)

4. 출력값은?
① 1
② 5
③ 6
④ none
⑤ error발생

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

x = 5
x_is_one()
print(x)
```

5. 출력값은?
① 3 5 7
② 3 5 7 9
③ 4 6 8
④ range(3, 9)
⑤ error 발생

```
for i in range(3, 9, 2):
    print(i)
```

6. 출력값
① 78
② 81
③ 85
④ 91
⑤ ERROR 발생

```
grades = [[62, 75, 77], [78, 81, 86], [85, 91, 89]]
print(grades[2][1])
```

4. (2) 5. (2) 6. (4)

7. 다른 결과값은
   ① print(not 4 < 3)
   ② print(false or true)
   ③ print(3 >= 2)
   ④ print((4 < 3) and 7 == 7)
   ⑤ print(not not true)

8. 9.0 출력 아닌것은?
   ① print(3 ** 2.0)
   ② print(float(3 * 4 -1)))
   ③ print(float(int(39/4)))
   ④ print(int("5") + float("4"))
   ⑤ print(str(3.0) * 3)

9. 출력값?

```
def f(x):
    print("f 시작")
    return x
    print("f 끝")

def g(x):
    print("g 시작")
    return x * x
    print("g 끝")

print(f(2))
print(g(3))
```

7. (4) 8. (5) 9. f 시작
2
g 시작

문제

1. 레시피 출력

    1. 준비된 컵에 초코 소스를 넣는다.
    2. 에스프레소를 추출하고 잔에 부어 준다.
    3. 초코 소스와 커피를 잘 섞어 준다.
    4. 거품기로 우유 거품을 내고, 잔에 부어 준다.
    5. 생크림을 얹어 준다.

코드 & 설명

```python
def cafe_mocha_recipe():
    print("1. 준비된 컵에 초코 소스를 넣는다")
    print("2. 에스프레소를 추출하고 잔에 부어 준다.")
    print("3. 초코 소스와 커피를 잘 섞어 준다.")
    print("4. 거품기로 우유 거품을 내고, 잔에 부어 준다.")
    print("5. 생크림을 얹어 준다.")
# 테스트 코드
cafe_mocha_recipe()
cafe_mocha_recipe()
```

함수 정의
- def 에 (, ,) 은 파라미터라고 부르며 input 값
- Index라고 부르는 명칭은 흔히 열 column을 지칭

함수 실행
- 함수 이름 ()

문제

1. 출력

1시간에 5달러 벌었습니다.
5시간에 25달러 벌었습니다.
1시간에 5710.8원 벌었습니다.
5시간에 28554.0원 벌었습니다.

코드 & 설명

```python
wage = 5  # 시급 (1시간에 5달러)
exchange_rate = 1142.16  # 환율 (1달러에 1142.16원)

# "1시간에 5달러 벌었습니다." 출력
print("{}시간에 {}{} 벌었습니다.".format(1, wage * 1, "달러"))

# "5시간에 25달러 벌었습니다." 출력
print("{}시간에 {}{} 벌었습니다.".format(5, wage * 5, "달러"))  # 코드를 채워

# "1시간에 5710.8원 벌었습니다." 출력
print("{}시간에 {}{} 벌었습니다.".format(1, wage * exchange_rate * 1, "원"))

# "5시간에 28554.0원 벌었습니다." 출력
print("{}시간에 {}{} 벌었습니다.".format(5, wage * exchange_rate * 5, "원"))
```

Format 영역을 method라고 부르며 function 이라고 이해함

문제

1. 홀인지 짝인지?.

짝수인 경우, 즉 number가 2로 나누어 떨어질 경우에는 True를 리턴해 줍니다.
홀수인 경우, 즉 number가 2로 나누어 떨어지지 않을 경우에는 False를 리턴해 줍니다.

코드 & 설명

```python
def is_evenly_divisible(number):
    return number % 2 == 0
    # 코드를 작성하세요


# 테스트
print(is_evenly_divisible(3))
print(is_evenly_divisible(7))
print(is_evenly_divisible(8))
print(is_evenly_divisible(218))
print(is_evenly_divisible(317))
```

def 가 참이 면 true임
% 나누고 남는 값

문제

1. 거스름돈 계산기

예를 들어 33,000원짜리 물건을 사기 위해 100,000원을 냈다면,

50,000원 1장
10,000원 1장
5,000원 1장
1,000원 2장
이런 식으로 '가장 적은 수'의 지폐를 거슬러 주는 것입니다.

몇 장을 거슬러 줘야 할까?

67,000원을 거슬러 줘야 하면, 50,000원 지폐는 몇 장 주면 될까요? 67,000원에 50,000원이 몇 번 들어가는지 확인하면 되죠? 파이썬에서는 버림 나눗셈(//)을 사용하면 이를 알 수 있습니다.

```
change // 50000  # 50,000원 지폐 개수
```

거슬러 주고 얼마가 남았을까?

67,000원에서 50,000원으로 최대한 거슬러 주고 남은 금액은 17,000원입니다. 파이썬에서는 나머지 연산(%)을 사용하면 이를 알 수 있습니다.

```
change % 50000  # 50,000원 지폐로 거슬러 주고 남은 금액
```

만약 50,000원과 10,000원을 최대한 거슬러 주고 남은 금액은 뭘까요? 단순하게 생각하면 `change % 50000 % 10000`인데요. 조금만 머리를 굴려 보면 이게 `change % 10000`과 같다는 걸 알 수 있습니다. 50,000은 10,000의 배수이기 때문이죠!

그럼 50,000원, 10,000원, 5,000원을 최대한 거슬러 주고 남은 금액은 어떻게 계산할까요? 단순하게 생각하면 `change % 50000 % 10000 % 5000`이지만, 그냥 간단하게 `change % 5000`만 해도 똑같은 결과가 나옵니다. 50,000과 10,000은 둘 다 5,000의 배수이기 때문입니다!

코드 & 설명

```python
def calculate_change(payment, cost):
    change = payment - cost

    fifth_count = change // 50000
    ten_count = (change % 50000) // 10000
    five_count = (change % 10000) // 5000
    one_count = (change % 5000) // 1000

    print("50000원 지폐 :{}장".format(fifth_count))
    print("10000원 지폐 :{}장".format(ten_count))
    print("5000원 지폐 :{}장".format(five_count))
    print("1000원 지폐 :{}장".format(one_count))


calculate_change(100000, 33000)
print()
calculate_change(500000, 378000)
```

문제

코드 & 설명

1. while

```
1  i = 2
2  while i <= 100 :
3      print(i)
4      i = i + 2
5
```

시작 변수가 밖에 있음

2. While

100 이상의 자연수 중 가장 작은 23의 배수를 출력해 보세요

```
i = 100

while (i % 23) != 0 :
    i = i + 1

print(i)
```

I 를 23으로 나눈 남은 값이 0이 아니면 참이니 +1을 더해라
반복하다 남는 값이 0이 된수면 반복문 중단 하고 I 값을프린터하라

문제

1. 학점계산기

50점 만점의 중간고사와 50점 만점의 기말고사
A: 90점 이상
B: 80점 이상 90점 미만
C: 70점 이상 80점 미만
D: 60점 이상 70점 미만
F: 60점 미만

코드 & 설명

```python
def print_grade(midterm_score, final_score):
    total = midterm_score + final_score
    if total >= 90 :
        print("A")
    elif total >= 80 :
        print("B")
    elif total >= 70 :
        print("C")
    elif total >= 60:
        print("D")
    elif total < 60:
        print("F")
    # 코드를 쓰세요.


# 테스트
print_grade(40, 45)
print_grade(20, 35)
print_grade(30, 32)
print_grade(50, 45)
```

문제

1. While과 if 문 조합

100 이하의 자연수 중 8의 배수이지만 12의 배수는 아닌 것을 모두 출력

코드 & 설명

```
1  i = 1
2
3  while i <= 100 :
4      if i % 8 == 0 and i % 12 != 0:
5          print (i)
6      i = i + 1
```

문제

1. While과 if 문 조합

1,000보다 작은 자연수 중 2 또는 3의 배수의 합

코드 & 설명

```python
i = 1
total = 0

while i < 1000:
    if i % 2 == 0 or i % 3 == 0:
        total += i
    i = i + 1

print(total)

```

I = 1 total =0

I가 1000보다 작으면,
I가 2 or 3으로 나눈 나머지값이 0이면 total 에 I 값을 더해라
i=1 은 나머지값이 0아 아니니 그냥 사라지며 i+1이 if 값에 들어감. 나머지 값이 0이니 total 값에 2 카운터. 999까지 반복.

Total이 밖에 나오면 if문 성립이 안됨.

문제

1. 약수 찾기

정수 120의 약수를 모두 출력하고, 총 몇개의 약수가 있는지 출력

코드 & 설명

```python
i = 1
n = 120
count = 0

while i <= 120:
    if n % i == 0:
        print(i)
        count += 1
    i += 1

print("{}의 약수는 총 {}개입니다.".format(n, count))
```

I 가 120보다 작으면,
만약, n/I 나머지가 0이면
I값을 프린터
카운터에 1더하고
i+1로 와일 구문 반복 ㄱㄱㄱ
I가 120이 될 때 까지

문제

1. 약수 찾기

1988년 쌍문동에 사는 택이는 바둑 대회 우승 상금으로 5,000만원을 받았습니다. 하지만 바둑 외에는 아는 게 없으니, 이웃 어른들에게 이 돈으로 무엇을 해야 할지 물어보기로 하였습니다.

은행에서 근무하는 동일 아저씨는 은행에 돈을 맡겨서 매년 이자로 12%씩 받는 것을 추천하셨습니다. 1년 후인 1989년에는 5,000만원의 12% 이자인 600만원이 더해져 5,600만원이 된다고 하면서요.

이 이야기를 들은 미란 아주머니는 고작 12% 때문에 생돈을 은행에 넣느냐며, 얼마 전 지어진 은마아파트를 사라고 추천하셨습니다. 당시 은마아파트의 매매가는 5,000만원이었죠.

2016년 기준 은마아파트의 매매가는 11억원인데요. 1988년 은행에 5,000만원을 넣었을 경우 2016년에는 얼마가 있을지 계산하여, 동일 아저씨와 미란 아주머니 중 누구의 말을 듣는 것이 좋았을지 판단해 보세요. 2016년 은행에 얼마가 있을지는 꼭 while 문을 사용해서 계산해 주세요!

2016년에 은행에 저축해 둔 금액이 더 크면, *원 차이로 동일 아저씨 말씀이 맞습니다.가 출력되도록 하세요. 반대로 은마아파트의 가격이 더 크면, *원 차이로 미란 아주머니 말씀이 맞습니다.가 출력되도록 하세요. 여기서는 꼭 if문을 사용해 주세요!

코드 & 설명

```python
INTEREST_RATE = 0.12
APARTMENT_PRICE_2016 = 1100000000

year = 1988
bank_balance = 50000000

while year < 2016:
    bank_balance = bank_balance * (1 + INTEREST_RATE)
    year += 1

if bank_balance > APARTMENT_PRICE_2016:
    print("{}원 차이로 동일 아저씨 말씀이 맞습니다.".format(int(bank_balance - APARTMENT_PRICE_2016)))
else:
    print("{}원 차이로 미란 아주머니 말씀이 맞습니다.".format(int(APARTMENT_PRICE_2016 - bank_balance)))

```

상수
이자율 (INTEREST_RATE) → 12%로 고정
2016년 은마아파트 가격 (APARTMENT_PRICE_2016) → 11억원으로 고정

변수
연도 (year) → 1988부터 2016까지 바뀜
은행 잔액 (bank_balance) → 50000000으로 시작해서 매년 쌓임

반복문을 이용해서 1988년 부터 2016까지 얼마나 쌓이는지 계산 ㄱㄱㄱ

문제

1. 피보나치 수열

피보나치 수열의 1번 항과 2번 항은 각각 1입니다.
3번 항부터는 바로 앞 두 항의 합으로 계산

코드 & 설명

```
previous = 0
current = 1
i = 1

while i <= 50:
    print(current)
    temp = previous  # previous
    previous = current
    current = current + temp  #
    i += 1
```

문제

코드 & 설명

1. 구구단 (while문)

```python
i = 1
while i <= 9:
    j = 1
    while j <= 9:
        print("{} * {} = {}".format(i, j, i * j))
        j += 1
    i += 1
```

문제

코드 & 설명

1. 구구단 (while문)

```python
i = 1
while i <= 9:
    j = 1
    while j <= 9:
        print("{} * {} = {}".format(i, j, i * j))
        j += 1
    i += 1
```

1. 구구단 (for 문)

```python
for i in range(1, 10):
    for j in range(1, 10):
        print("%d * %d = %d" % (i, j, i * j))
```

문제

1. 온도 단위 바꾸기

코드 & 설명

```python
def fahrenheit_to_celsius(fahrenheit):
    return (fahrenheit - 32) * 5 / 9


temperature_list = [40, 15, 32, 64, -4, 11]
print("화씨 온도 리스트: " + str(temperature_list))  # 화씨 온도 출력

# 리스트의 값들을 화씨에서 섭씨로 변환하는 코드
i = 0
while i < len(temperature_list):
    temperature_list[i] = round(fahrenheit_to_celsius(temperature_list[i]), 1)
    i += 1
print("섭씨 온도 리스트: {}".format(temperature_list))  # 섭씨 온도 출력
```

Print 안에 있으면 while 문 돌때마다 프린트함. 거짓이면 while문 빠져 나옴

문제

코드 & 설명

1. 환전하기

```python
def krw_to_usd(krw):
    return krw / 1000

def usd_to_jpy(usd):
    return usd / 8 * 1000

prices = [34000, 13000, 5000, 21000, 1000, 2000, 8000, 3000]
print("한국 화폐 : {}".format(prices))

i = 0
while i < len(prices):
    prices[i] = krw_to_usd(prices[i])
    i += 1
print("미국 화폐 : {}".format(prices))

i = 0
while i < len(prices):
    prices[i] = usd_to_jpy(prices[i])
    i += 1
print("일본 화폐 : {}".format(prices))
```

실행 채점

</> 실행 결과

```
한국 화폐 : [34000, 13000, 5000, 21000, 1000, 2000, 8000, 3000]
미국 화폐 : [34.0, 13.0, 5.0, 21.0, 1.0, 2.0, 8.0, 3.0]
일본 화폐 : [4250.0, 1625.0, 625.0, 2625.0, 125.0, 250.0, 1000.0, 375.0]
```

(krw)에 비어있으면 문제가?

prices[0]은
리스트에 첫번째 값

문제

1. 환전하기

1.numbers라는 빈 리스트를 만들고 리스트를 출력한다.
2.append를 이용해서 numbers에 1, 7, 3, 6, 5, 2, 13, 14를 순서대로 추가한다. 그 후 리스트를 출력한다.
3.numbers 리스트의 원소들 중 홀수는 모두 제거한다. 그 후 다시 리스트를 출력한다.
4.numbers 리스트의 인덱스 0 자리에 20이라는 수를 삽입한 후 출력한다.
5.numbers 리스트를 정렬한 후 출력한다

코드 & 설명

```python
numbers = []
print(numbers)

numbers.append(1)
numbers.append(7)
numbers.append(3)
numbers.append(6)
numbers.append(5)
numbers.append(2)
numbers.append(13)
numbers.append(14)

print(numbers)

i = 0
while i < len(numbers):
    if numbers[i] % 2 == 1:
        del numbers[i]
    else:
        i += 1
print (numbers)

numbers.insert(0,20)
print(numbers)

numbers.sort()
print(numbers)
```

Append 는 1개만 넣을수 있음

문제

코드 & 설명

1. numbers의 인덱스와 원소

```
numbers = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31]

# 인덱스와 원소 출력
for i in range(len(numbers)):
    print(i, numbers[i])
```

Len(numbers) 가 11개이니 0~910개 인수에서 I 가 하나씩 가져간다
Numners[0] = 2이다
0, 2 -> 1, 3 -> ㅇㅇㅇ

```
def print_squre(x):
    print(x * x)

print(print_squre(3))
```

9
None 출력

Print(print(3)) 는 없는값 이니

```
for i in range(11):
    print("{}^{} = {}".format(2, i, 2 ** i))
```

거듭제곱 **

문제

코드 & 설명

1. 퓌타고라스

```python
for a in range(1, 1000):
    for b in range(1, 1000):
        c = 1000 - a - b
        if a * a + b * b == c * c and a < b < c:
            print(a * b * c)
```

문제

코드 & 설명

1. 리스트 뒤집기

```python
numbers = [2, 3, 5, 7, 11, 13, 17, 19]

# 리스트 뒤집기
for left in range(len(numbers) // 2):
    # 인덱스 left와 대칭인 인덱스 right 계산
    right = len(numbers) - left - 1

    # 위치 바꾸기
    numbers[right], numbers[left] = numbers[left], numbers[right]

print("뒤집어진 리스트: " + str(numbers))
```

```python
numbers = [2, 3, 5, 7, 11, 13, 17, 19]

numbers.sort(reverse):
print("뒤집어진 리스트:{}".format(numbers))
# 리스트 뒤집기
# 코드
```

이런건 없나?

문제

코드 & 설명

1. 단어장 만들고 추가하기

```python
# 1. 단어장 만들기
vocab = {
    'sanitizer': '살균제',
    'ambition': '야망',
    'conscience': '양심',
    'civilization': '문명'
}
print(vocab)


# 2. 새로운 단어들 추가
vocab['privilege'] = '특권'
vocab['principle'] = '원칙'
print(vocab)
```

문제

코드 & 설명

1. 사전 뒤집기

```python
# 언어 사전의 단어와 뜻을 서로 바꿔주는 함수
def reverse_dict(dict):
    new_dict = {}  # 새로운 사전

    for key, value in dict.items():
        new_dict[value] = key
    # dict의 key와 value를 뒤집어서 new_dict에 저장
    # 코드를 입력하세요.

    return new_dict  # 변환한 새로운 사전 리턴


# 영-한 단어장
vocab = {
    'sanitizer': '살균제',
    'ambition': '야망',
    'conscience': '양심',
    'civilization': '문명',
    'privilege': '특권',
    'principles': '원칙'
}

# 기존 단어장 출력
print("영-한 단어장\n{}\n".format(vocab))

# 변환된 단어장 출력
reversed_vocab = reverse_dict(vocab)
print("한-영 단어장\n{}".format(reversed_vocab))
```

Dict은 어디서 오는 파라미터인가?
Dcit.item 파라미터, . 메쏘드 형태로 키와 벨류를 같이 리턴하는 기능

문제

코드 & 설명

1. 사전 뒤집기

```python
# 투표 결과 리스트
votes = ['김영자', '강승기', '최만수', '김영자', '강승기', '강승기', '최만수', '김영자', \
'최만수', '김영자', '최만수', '김영자', '김영자', '최만수', '최만수', '최만수', '강승기', \
'강승기', '김영자', '김영자', '최만수', '김영자', '김영자', '강승기', '김영자']

# 후보별 득표수 사전
vote_counter = {}

# 리스트 votes를 이용해서 사전 vote_counter를 정리하기
for name in votes:
    if name not in vote_counter:
        vote_counter[name] = 1
    else:
        vote_counter[name] += 1
    # 코드를 작성하세요.

# 후보별 득표수 출력
print(vote_counter)
```

{} 딕스너리 쌍 데이터 ?
[]다양한 인수 넣는 보트 카운트에
빈 리스트 만들기

네임변수가 보트 리스트를 반복함
만약 네임변수가 보트 카운트에 없으면 1
아니면 다음 +1

{김영자 : 11, 강승기 :6}로 출력

문제

1. 자릿수 합 구하기

sum_digit 함수를 정의하기 위한 힌트를 드리겠습니다.
자릿수 합을 보관하는 변수 total을 0으로 정의한다.
정수형 num을 문자열로 바꾼다.
문자열은 리스트와 유사하다는 점을 이용하여, 반복적으로 각 자릿수를 받는다.
각 자릿수를 정수형으로 변`환한다.
각 자릿수를 total에 더한다.
반복문을 이용해서 total을 모두 계산한 후, total을 리턴한다.

코드 & 설명

```
def sum_digit(num):
    total = 0
    str_num = str(num)

    for digit in str_num:
        total += int(digit)

    return total


# sum_digit(1)부터 sum_digit(1000)
digit_total = 0
for i in range(1, 1001):
    digit_total += sum_digit(i)
```

Num 변수를 가진 sum_digit함수를 정의할께

Total은 0 이며, num 변수를 분자열로 바꾼걸 str_num이라고 함

str_num 문자열 안에 변수를 찾아서 digit이라고 하고

숫자열로 바뀐 digit 변수를 더해서 total 에 저장함

str_num 다돌면 합친 total 값은 리턴해노음

Digital_total 은 0

1부터 1000까지 를 I 변수로 순차적으로 반복함

Sum_digit(1)~(1000)까지 digit_totla에 쌓임

문제

코드 & 설명

1. 주민등록번호 가리기

문자열은 수정이 불가능합니다. 하지만 문자열과 유사한 리스트는 수정이 가능하죠? 그러면 문자열 security_number를 리스트로 변환한 후, 마지막 네 원소를 '*'로 바꿔 주면 됩니다. 그리고 나서 그 리스트를 다시 하나의 문자열로 합치면 되겠죠?

```python
def mask_security_number(security_number) :
    security_list = list(security_number)

    security_str = ""
    for i in range(len(security_list)) :
        if len(security_list)-4 <= i :
            security_list[i] = "*"
        security_str += security_list[i]


    return(security_str)
# 테스트
print(mask_security_number("880720-1234567"))
print(mask_security_number("8807201234567"))
print(mask_security_number("930124-7654321"))
print(mask_security_number("9301247654321"))
print(mask_security_number("761214-2357111"))
print(mask_security_number("7612142357111"))
```

문제

코드 & 설명

1. 펠린드롬

토마토"나 "기러기"처럼 거꾸로 읽어도 똑같은 단어를 '팰린드롬(palindrome)'이라고 부릅니다

```python
def is_palindrome(word):
    for left in range(len(word) // 2):
        # 한 쌍이라도 일치하지 않으면 바
        right = len(word) - left - 1
        if word[left] != word[right]:
            return False

    # for문에서 나왔다면 모든 쌍이 일치
    return True

# 테스트
print(is_palindrome("racecar"))
print(is_palindrome("stars"))
print(is_palindrome("토마토"))
print(is_palindrome("kayak"))
print(is_palindrome("hello"))
```

문제

1. 숫자 맞히기 게임

상수

게임의 정답 → ANSWER
총 기회 수 → NUM_TRIES

코드 & 설명

```python
import random

# 상수 정의
ANSWER = random.randint(1, 20)
NUM_TRIES = 4

# 변수 정의
guess = -1
tries = 0

while guess != ANSWER and tries < NUM_TRIES:
    guess = int(input("기회가 {}번 남았습니다. 1-20 사이의 숫자를 맞혀보세요: ".format(NUM_T
    tries += 1

    if ANSWER > guess:
        print("Up")
    elif ANSWER < guess:
        print("Down")

if guess == ANSWER:
    print("축하합니다. {}번 만에 숫자를 맞히셨습니다.".format(tries))
else:
    print("아쉽습니다. 정답은 {}입니다.".format(ANSWER))
```

1~20사이 값만 아니면 됨

NUM_TRIES - tries

문제

1. **코딩에 빠진 닭**

2. **Txt. 파일 열어서 월평균 매출액 구하기**
   예)
   1일: 453400 2일: 388600 3일: 485300
   4일: 477900 5일: 432100 6일: 665300
   7일: 592500 8일: 465200 9일: 413200
   10일: 523000 11일: 488600 12일: 431500
   13일: 682300 14일: 633700 15일: 482300
   16일: 391400 17일: 512500 18일: 488900
   19일: 434500 20일: 645200 21일: 599200
   22일: 472400 23일: 469100 24일: 381400
   25일: 425800 26일: 512900 27일: 723000
   28일: 613600 29일: 416700 30일: 385600
   31일: 472300

코드 & 설명

```python
with open('data/chicken.txt', 'r') as f:
    total_revenue = 0
    total_days = 0

    for line in f:
        data = line.strip().split(": ")
        revenue = int(data[1])

        total_revenue += revenue
        total_days += 1

    print(total_revenue / total_days)
```

R은 리드

Data 는 [1일, 4560000]
[2일, 567000] 형태 리턴

Data에 2번째 인수를 정수로~

문제

코드 & 설명

1. 단어장 만들기

이 프로그램은 콘솔로 영어 단어와 한국어 뜻을 받고, vocabulary.txt라는 새로운 텍스트 파일에 단어와 뜻을 정리하는데요. 사용자가 새로운 단어와 뜻을 입력할 때마다 vocabulary.txt에 작성되는 것입니다.
사용자는 반복적으로 단어와 뜻을 입력하는데, 단어나 뜻으로 q를 입력하는 순간 프로그램은 즉시 종료됩니다. 사용자가 q를 입력하고 나면 파일은 더 이상 바뀌지 않아야 합니다.

```python
with open('vocabulary.txt','w') as f:
    while true:
        english_word = input("영어 단어를 입력하세요: ")
        if english_word == "q":
            break

        korea_word = input("한국어 뜻을 입력하세요: ")
        if korea_word == "q":
            break

        f.write('{}: {}\n'.foemat(english_word, korea_word))

```

W wright

문제

1. 단어 퀴즈

이번에는 이 파일의 단어들을 가지고 학생들에게 문제를 내 주는 프로그램을 만들려고 합니다.

프로그램은 콘솔에 한국어 뜻을 알려 줄 것이고, 사용자는 그에 맞는 영어 단어를 입력해야 합니다. 사용자가 입력한 영어 단어가 정답이면 "맞았습니다!"라고 출력하고, 틀리면 "아쉽습니다. 정답은 OOO입니다."가 출력되어야 합니다.

문제를 내는 순서는 vocabulary.txt에 정리된 순서입니다.

코드 & 설명

```
1  with open('vocabulary.txt', 'r', encoding='utf-8') as f:
2
3      for line in f:
4          data = line.strip().split(": ")
5          english_word, korea_word = data[0], data[1]
6
7      guess = input("{}: ".format(korea_word))
8
9      if guess == english_word:
10         print("정답입니다.\n")
11     else:
12         print("아쉽습니다. 정답은{}입니다.\n".format(english_word))
13
```

1. 파일을 열자

파일을 한 줄 씩 읽자

Strip으로 line에서 "₩n"제거
Split으로 영어단어와 한국단어 나누기
₩인덱스 0은 영어단더,1은 한국어 지정

문제

코드 & 설명

1. 고급 단어퀴즈

vocabulary.txt에서 랜덤한 순서로 나오게 하기

```python
import random

vocab = {}
with open('vocabulary.txt', 'r') as f:
    for line in f:
        data = line.strip().split(": ")
        english_word, korean_word = data[0], data[1]
        vocab[english_word] = korean_word

while true:
    keys = list(vocab.keys())
    index = random.randint(0, len(keys) - 1)
    english_word = keys[index]
    korean_word = vocab[english_word]

    guess = input("{}: ".format(korea_word))

    if guess == "q":
        break

```

키, 밸류 딕셔너리 형태로 만들겠다는 문법이다
Vocab 딕셔너리에 txt 내용을 추가

Korea_word 는 vocab 딕셔러니 안에 key에 있는 value을 가져와서 korea word에 넣는다

main.py

```python
def count_matching_numbers(numbers, winning_numbers):
    count = 0

    for num in numbers:
        if num in winning_numbers:
            count = count + 1

    return count


def check(numbers, winning_numbers):
    count = count_matching_numbers(numbers, winning_numbers[:6])
    bonus_count = count_matching_numbers(numbers, winning_numbers[6:])
    # 코드를 작성하세요.


# 테스트
print(check([2, 4, 11, 14, 25, 40], [4, 12, 14, 28, 40, 41, 6]))
print(check([2, 4, 11, 14, 25, 40], [2, 4, 10, 11, 14, 40, 25]))
```

Count는맞은 갯수다

main.py

```python
from random import randint


def generate_numbers(n):
    numbers = []

    while len(numbers) < n:
        new_number = randint(1, 45)
        if new_number not in numbers:
            numbers.append(new_number)

    return numbers


def draw_winning_numbers():
    winning_numbers = generate_numbers(7)
    return sorted(winning_numbers[:6]) + winning_numbers[6:]


def count_matching_numbers(numbers, winning_numbers):
    count = 0

    for num in numbers:
        if num in winning_numbers:
            count = count + 1

    return count
```

```python

def check(numbers, winning_numbers):
    count = count_matching_numbers(numbers, winning_numbers[:6])
    bonus_count = count_matching_numbers(numbers, winning_numbers[6:])

    if count == 6:
        return 1000000000
    elif count == 5 and bonus_count == 1:
        return 50000000
    elif count == 5:
        return 1000000
    elif count == 4:
        return 50000
    elif count == 3:
        return 5000
    else:
        return 0
관련 질문
```