

데이터분석 프로그래밍 중간고사

다음의 문제를 풀어서 .ipynb 파일과 .pdf 파일을 아주 비비에 올리세요.

강의자료, 구글검색, 책 등을 활용할 수 있는 오픈북 문제입니다.

학번:

이름:김효륜

다음의 문제를 읽고 해당 코드를 작성하시오. (하나당 10점)

1. 다음 소스 코드를 완성하여 확장자가 .jpg, .gif인 이미지 파일만 출력되게 만드세요.

```
In [1]: # 이 부분은 건드리지 말것
files = ['font', 'jpg.xls', '2.png', 'app.gif', '1.png', '10.jpg', '11.gif', '2.jpg', '3.png', 'table.xlsx', 'spec.docx']

# 풀이
print(list(filter(lambda x: x.find('.jpg')!=-1 or x.find('.gif')!=-1,files)))

['app.gif', '10.jpg', '11.gif', '2.jpg']
```

2. 다음 소스 코드를 완성하여 리스트 a에 들어있는 문자열 중에서 길이가 4이상인 것들중 리스트 형태로 출력되게 만드세요(리스트 표현식 사용). 그런 후 중복된 문자열을 제거하여 한번만 출력되게 하세요 (set 사용). 결과 출력시 해당 문자열이 몇개 있는지 len() 함수를 이용하여 함께 출력해주세요.

```
In [2]: # 이 부분은 건드리지 말것
a = ['alpha', 'bravo', 'charlie', 'delta', 'echo', 'foxtrot', 'golf', 'hotel', 'india', 'al', 'alph', 'alpha', 'char', 'charlie']

# 풀이
b=[i for i in a if len(i)>=4]
print(set(b))

{'char', 'bravo', 'alph', 'india', 'hotel', 'golf', 'delta', 'echo', 'alpha', 'charlie', 'foxtrot'}
```

```
In [ ]:
```

```
In [ ]:
```

3. 다음 소스 코드를 완성하여 파일 경로에서 파일명과 파일명 바로 앞에 있는 폴더까지 포함하여 출력되게 만드세요. 단, 경로에서 폴더의 깊이가 달라지더라도 파일명만 출력할 수 있어야 합니다. 결과는 다음과 같습니다.

```
['jykang', 'python.exe']
```

```
In [7]: # 이 부분은 건드리지 말것
path = 'C:WWUsersWWjygangWWpython.exe'
## 풀이
x=path.split('WW')
x.reverse()
filename=x[0:2]

# 이 부분은 건드리지 말것
print(filename)

['python.exe', 'jygang']
```

In []:

In []:

In []:

In []:

4. 문자열을 입력받아 각 글자 중 대문자인 경우만 빈도수를 세어보시오. (letter.isupper() 사용)

```
In [11]: ## 이 부분은 건드리지 말것
string=input("문자열을 입력하시오: ")
countTable={}
## 풀이
for letter in string:
    if letter.isupper():
        countTable[letter] = countTable.get(letter, 0) + 1

print(countTable)

문자열을 입력하시오: Hello World
{'H': 1, 'W': 1}
```

In []:

In []:

In []:

In []:

5. 문자열 안에 있는 문자의 개수, 대문자의 개수, 소문자의 개수를 계산하는 프로그램을 작성하여 보자.

실행결과예

문자열을 입력하시오: A picture is WORTH a thousand words.
{'alphas': 29, 'uppers': 6, 'lowers': 23}

```
In [13]: ## 이 부분은 건드리지 말것
sentence=input("문자열을 입력하시오: ")
table={'alphas':0, "uppers":0, "lowers":0}

## 풀이
for i in sentence:
    if i.isalpha():
        table["alphas"]+=1
    if i.isupper():
        table["uppers"]+=1
    if i.islower():
        table["lowers"]+=1
print(table)
```

문자열을 입력하시오: A picture is WORTH a thousand words.
{'alphas': 29, 'uppers': 6, 'lowers': 23}

In []:

In []:

6. text에 들어가 있는 문자들 중 모든 문자가 소문자인 'to'만 찾으면 되며 'too', 'ton', 등은 포함하지 않아야 합니다.

```
In [15]: text = '''the grown-ups' response, this time, was
too advise me to lay aside my drawings of boa constrictors,
whether from the inside or the outside, and devote myself
instead to geography, history, arithmetic, and grammar.
That is why, at the, age of six, I gave up what might have
been a magnificent career as a painter. I had been disheartened
by the failure of my Drawing Number One and my Drawing Number Two.
Grown-ups never understand anything by themselves, and it is tiresome
for children to be always and forever explaining things to the.'''
cnt = 0
for word in text.split():
    if word.strip(',.')== 'to':
        cnt+=1
print(cnt)
```

4

In []:

In []:

7. 주어진 단어의 문자열 갯수를 세어보시오. (hint: word.split() 이용, dictionary comprehension 사용)

실행결과 예 {'Apple': 2, 'Black': 1, 'Orange': 1, 'Pear': 1}

```
In [18]: word = "Apple Black Orange Apple Pear"
word=word.split()
d={letter:word.count(letter) for letter in word}
print(d)

{'Apple': 2, 'Black': 1, 'Orange': 1, 'Pear': 1}
```

In []:

In []:

In []:

8. list comprehension을 사용하여 주어진 dictionary에서 email 부분만 추출하여 소문자로 변경하시오.

hint) d.items() 대신 d.values() 사용

실행예: ['john@gmail.com', 'tom@gmail.com']

```
In [26]: d = {'JOHN' : 'JOHN@GMAIL.COM', 'tom':'tom@Gmail.com'}
emails = [value.lower() for value in d.values()]
print(emails)

['john@gmail.com', 'tom@gmail.com']
```

In []:

In []:

In []:

In []:

In []:

In []:

In []:

9. 문자열 두개를 입력받아 .split()을 이용하여 단어리스트로 만든 후 두 문자열을 zip()과 dict()를 사용하여 dictionary로 변환하시오.

실행예

문자열 두개를 입력하세요

문자열1: kang kim park

문자열2: 101 202 303

{'kang': '101', 'kim': '202', 'park': '303'}

```
In [27]: a=["kang", "kim", "park"]
          b=["101", "202", "303"]
          ab=zip(a,b)
          dab=dict(ab)
          print(dab)

{'kang': '101', 'kim': '202', 'park': '303'}
```

In []:

10. 다음의 문제를 읽고 지시대로 푸세요 (10점)

연락처

- 윤승식 : 010-3348-2859
- 허준 : 010-7708-5617
- 신민철 : 010-4729-8241
- 강승식: 010-2323-1122

요구사항

1. 위에 제시된 연락처만 이용하여 적절한 자료형을 이용하여 만들어주세요.(이름 중복 등은 고려하지 않도록 합니다)
2. 연락처에 기록된 전화번호 검색하는 contact_search() 함수 내부를 list comprehension 방식을 이용하여 만들어주세요. 함수의 결과물로 전화번호를 리턴 할 수 있도록 해주세요
3. 해당 함수를 통해 연락처를 검색할 때, 이름의 일부만 입력해도 검색이 가능하도록 해주세요.

예시: 준 -> 010-7708-5617, 신민철 -> 010-4729-8241, 윤 -> 010-3348-2859

```
In [29]: phone = {"윤승식": "010-3348-2859", "허준": "010-7708-5617", "신민철": "010-4729-8241", "강승식": "010-2323-1122"}
def contact_search(txt, phone):
    ans = []
    for key, value in phone.items():
        if txt in key:
            ans.append(value)
    return ans
print(contact_search("준", phone))
print(contact_search("신민철", phone))
print(contact_search("윤", phone))

['010-7708-5617']
['010-4729-8241']
['010-3348-2859']
```

11. 스키테일 암호 문제 (15점)



고대 그리스의 역사학자 플루타르크에 따르면 약 2,500년 그리스 지역의 옛 나라인 스파르타에서는 전쟁터에 나가있는 군대에 비밀메시지를 전할 때 암호를 그때 당시로서는 아무나 쉽게 암호문을 열어 볼수 없는, 아주 교묘하고도 획기적인 방법이었다. 이때 사용한 암호화방법은 스키타일(Scytale)암호라 하고 당시의 전쟁터에선 다음과 같은 방법으로 암호화하였다.

1. 전쟁터에 나갈 군대와 본국에 남아있는 정부는 각자, 스키타일(Scytale)이라고 하는 굵기의 원통형 막대기를 나누어 갖는다.
2. 비밀리에 보내야 할 메시지가 생기면, 본국 정부의 암호 담당자는 스키타일에 가느다란 양피지 리본을 위에서 아래로 감은 다음 옆으로 메시지를 적는다.
3. 리본을 풀어내어 펼치면 메시지의 내용은 아무나 읽을 수 없게 된다..
4. 전쟁터에 나가있는 오로지 같은 굵기의 원통막대기를 가진 사람만이 메시지를 읽을 수 있다.

스키테일 암호를 만드는 방법은 간단합니다. 메시지의 간격(n)을 설정하고, 그 사이에 의미없는 문자열을 삽입하면 되는 것이죠.

윤승식 연구원은 여자친구와의 비밀 메시지를 위해 메시지의 간격을 설정하고, 스키테일 암호를 이용하기로 결정하였습니다.

자, 이제 윤승식 연구원을 위해 간격을 정해주면 스키테일 암호문을 만들어내는 함수 encrypt()와, 이런 암호문을 해독하는 함수 decrypt()를 작성해 봅시다

요구사항

1. 문자열과, 막대기의 굵기를 입력받아 스키테일 암호문을 암호화하는 함수 `encrypt(string, n)`과, 스키테일 암호문을 풀어내는 함수 `decrypt(string, n)`을 작성하세요.
2. 사이에 집어넣을 임의의 문자열은 `rand_str` 함수를 이용하여 생성해주세요.
3. 암호화를 할 문자열과 메시지 간격을 사용자로 부터 입력받으세요. (hint> 메시지 간격의 경우 `int()` 함수를 이용하여 숫자로 바꾸세요)
4. 함수의 결과물로 `string`을 `return`해주세요.

<실행예>

```
암호화할 문자열:hello world
메시지 간격:2
hjgefmlsFlilozP MQwQooR0ryKlhjdFS
hlwl
```

```
In [30]: # 이부분은 건드리지 말것, 한번 실행하고 문제 풀기
import random

character = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
def rand_str(length, characters=character):
    """
    rand_str(length, characters)
    characters에 지정되어 있는 문자열을 임의로 추출하여 length 길이만큼 추출해줍니다.
    characters를 지정하지 않을 시 기본값으로 영어 알파벳 대,소문자에서 임의의 문자를 추출합니다.
    """
    return "".join(random.choices(characters, k=length))
```

```
In [ ]: rand_str(30)
```

```
In [31]: #encrypt()
def encrypt(string,n):
    clean_string = ''.join(string.split()).lower()
    message=""
    for i in range(0,len(string)):
        message+=string[i]
        message+=rand_str(n)
    return message
#encrypt()
def decrypt(string,n):
    message=""
    for i in range(0,len(string),n+1):
        message+=string[i]
    return message
enc=encrypt("hello world",2)
dec=decrypt(enc,2)
print(enc)
print(dec)
```

```
hYNeZGIYGIvvoaT LjwGLozarqPlvtdct
hello world
```

```
In [ ]:
```

```
In [ ]:
```


12. 3행시 문제내기 (15점)

예를 들어 "강주영"을 입력했을 때 3행시를 잘 짓는지를 테스트하는 프로그램을 작성해 보세요.

강 : 강릉

주 : 주민은

영: 영동지방 사람입니다.

3행시를 넣으면(적절한 구조를 생각해서 파라미터로 넣으세요) 첫글자를 확인해서 "강주영"이 맞는지 확인하는 check3() 함수를 만들어주세요. 강주영이 맞다면 True, 아니면 False를 반환하도록 합니다. **이때 삼행시를 각각 두글자 이상 만들었는지도 함께 체크해주세요.** 그리고 한글자만 적었을 경우에 대한 적절한 메시지, 해당 첫글자로 시작하지 않았을때의 적절한 메시지도 함께 적어주세요.

위에 있는 시 말고, 다른 시를 만들어서 검사할 수 있도록 합니다. 우리 건용이는 소중하니까요! 실행예

강: 강릉
주: 주민
영: 영
두글자 이상으로 만들어주세요
False

강: 강릉
주: 민주
해당글자로 시작해야 합니다.
False

강: 강릉
주: 주민
영: 영동지방
True

```
In [32]: def check3(txt):  
          for i in txt:  
              first=input("%s: "%i)  
              if len(first) < 2:  
                  print("두 글자 이상으로 입력해주세요.")  
                  return False  
              elif first[0] != i:  
                  print("첫 글자로 시작해야 합니다.")  
                  return False  
          return True  
msg=input("삼행시를 만들 단어를 넣으세여:")  
check3(msg)
```

삼행시를 만들 단어를 넣으세여:강주영
강: 강릉
주: 주민
영: 영
두 글자 이상으로 입력해주세요.

Out[32]: False

```
In [34]: def check3(txt):
          for i in txt:
              first=input("%s: %i")
              if len(first) < 2:
                  print("두 글자 이상으로 입력해주세요.")
                  return False
              elif first[0] != i:
                  print("첫 글자로 시작해야 합니다.")
                  return False
          return True
          msg=input("삼행시를 만들 단어를 넣으세요:")
          check3(msg)
```

삼행시를 만들 단어를 넣으세요:강주영

강: 강릉

주: 주민

영: 영동지방

Out[34]: True

In []:

In []:

In []:

13. 보너스 문제, 자신이 하고 싶은 이야기에 대해 자유롭게 아는 문법을 사용하여 프로그램을 만들어 보세요 (20점)

In []: