1. Show the execution result of the following code snippets. (5*16=80 points)

(1)
```
my_dict = {'age': 21,'name': "kim", 'home':'Seoul'}
my_dict['grade'] = 3

d = []
for k, v in my_dict.items():
    if k=='name' or k=='grade':
        d.append(v)
print(d)
```
['Kim',3]

(2) Rewrite the code (1) with a single Python command line.
(hint: use the forms used in part (3))

d = [v for k,v in my_dict.items() if k == 'name' or k == 'grade']

(3)
```
A, B, C = range(6), ['a','b','c'], [(3,4),(7,24)]
a = [(x, x**2) for x in A if x % 2 == 0]
b = [(x,y) for x in A for y in B if x == 3]
c = [np.sqrt(x*x+y*y) for (x,y) in C]
print(a)
print(b)
print(c)
```
-> [(0,0),(2,4),(4,16)]
-> [(3,'a'),(3,'b'),(3,'c')]
-> [5,25]

(4)
```
import pandas as pd
df = pd.DataFrame(np.ones((3,4))).astype(int)
df.iloc[:2, 1] = 0
df.iloc[:1, 2] = 0
df.iloc[0,-1] = 0
print(df)
```
이건 풀이 쓰기 귀찮음

(5)
```
import numpy as np
data = np.array([[11,22,33],
                 [44,55,66],
                 [77,88,99]])
X, y = data[:, 1:], data[:, 0]
print(X)
print(y.reshape(-1,1))
```
22 33
55 66
88 99

11
44
77

(6)
```
from sklearn.feature_extraction.text import CountVectorizer
docs = ['This is the first document.',
        'Is this document the first document?',
        'No, this is the second.']
cv = CountVectorizer(lowercase=True)
words = cv.fit_transform(docs)
print('\t', cv.get_feature_names_out())
print(words.toarray())
```
['document' 'first' 'is' 'no' 'second' 'the' 'this']

[[1 1 1 0 0 1 1]
[2 1 1 0 0 1 1]
[0 0 1 1 1 1 1]]

(7)
```
from bs4 import BeautifulSoup
html_text = """
    <html> <body>
        <h1 id="title"> reading web page with python </h1>
        <p id="body"> page analysis </p>
        <td>some text</td>
        <ul> <li><a href = "http://www.naver.com"> naver</a></li>
             <li><a href = "http://www.daum.net"> daum</a></li>
        </ul>
</body> </html>
"""
soup = BeautifulSoup(html_text, 'html.parser')
print(soup.find_all('a'))
    [<a href="http://www.naver.com"> naver</a>,
     <a href="http://www.daum.net"> daum</a>]
for aa in soup.find_all('a'):
    print (aa.string)
```

naver, daum

(8)
```
import matplotlib.pyplot as plt
 x = np.linspace(0,1,50)
 y1 = np.sin(2*np.pi*x)
 y2 = np.cos(2*np.pi*x)

 fig, ax = plt.subplots(2,1)
 ax[0].plot(x, y1, 'r-*', lw=1)  # *-*-*
 ax[1].plot(x, y2, 'b--', lw=1)  # -----
```

(9)
```
with open("words.txt", "w") as f:
    f.write("Gildong HONG -  April Cross \n")
    f.write("CHULSOO Park - APRIL  CROSS \n")
    f.write("Taehee KIM - Champion")
names, votes = [], []
with open("words.txt", "r") as f:
    for line in f:
        line = line.strip()
        name, vote = line.split(" - ")
        name = name.strip().lower().replace("  "," ").replace(" ","-")
        vote = vote.strip().lower().replace("  "," ").replace(" ","-")
        names.append(name)
        votes.append(vote)
print(names)
print(votes)
```

names에는 “-” 기준 왼쪽거인데 gildong-hong 이런식
votes도 동일

(10)
```
import pandas as pd
df1 = pd.DataFrame([[1,2],[3,4]], index=['a','b'], columns=['A','B'])
df2 = pd.DataFrame([[3,3],[7,7]], index=['a','b'], columns=['X','Y'])
new_df = pd.concat([df1, df2], axis=1)
print(new_df)
```

axis 1쪽이 많아지게 그니까 columns가 A B X Y 가 되는거임

(11)
```
class Class1:
    def __init__(self, arr: list):
        self.arr1 = np.array(arr)
    def get_max(self):
        return(np.max(self.arr1)+ np.min(self.arr1))

class Class2(Class1):
    def __init__(self, arr: list):
        super().__init__(arr)
        self.arr2 = arr
    def get_argmax(self):
        return(np.argmax(self.arr2)+ np.argmin(self.arr2))

a1 = Class1([1, 4, 2, 3, 5, 6, 19])
a2 = Class2([5, 6, 9, 8, 3, 4, 11])
print(a1.get_max())
print(a2.get_argmax())

try:
    print(a2.get_max())
except:
    print("Error!")
```
20
10
14

(12)
```
from numpy import nan as NA
df = pd.DataFrame([[NA, 6.5, 3.],
                   [NA, NA, NA],
                   [NA, 6.5, 2.]])
print(df)
```
```
     0    1    2
0  NaN  6.5  3.0
1  NaN  NaN  NaN
2  NaN  6.5  2.0
```
```
print(df.fillna({0: 0.5, 1: -1, 2:df[2].mean()}))
```
0.5 6.5 3.0
0.5 -1 2.5
0.5 6.5 2.0

(13)
```
import numpy as np
N= 1000000
p = np.array([0, 0, 0, 0, 0, 0])

for i in range(N):
    k = np.random.randint(1,7)
    p[k-1] += 1

print((p/N).round(2))
```
이거 ㅈ댐 모르겠음

0.17,0.17,0.17,0.17,0.17,0.17

(14)
```
from sklearn.preprocessing import StandardScaler
X = np.array([1, 2, 1])
ss = StandardScaler()
X_sc = ss.fit_transform(X.reshape(-1,1))
print(X_sc.round(2))
```
x-평균 / sqrt(mean(편차^2))

(15)

```
from sklearn.preprocessing import LabelEncoder,OneHotEncoder

df = pd.DataFrame({'Age':[33,44,22,44,55,22],
                   'Gen':['Male','Female','Male','Female','Male','Male']})

le = LabelEncoder()
new_f1 = le.fit_transform(df.Gen)
ohe = OneHotEncoder()
new_f2 = ohe.fit_transform(df[['Gen']]).toarray()
print(le.classes_)
   ['Female' 'Male']
print(ohe.categories_)
   [array(['Female', 'Male'], dtype=object)]
#-------------
print(new_f1)
print(new_f2)
```

	Age	Gen
0	33	Male
1	44	Female
2	22	Male
3	44	Female
4	55	Male
5	22	Male

귀찮음 이슈 이건 레이블 인코딩 원 핫 인코딩 그냥 하셈

(16)

```
def myfunc(x, y):
    x += 100
    y.append(17)
    return

a, b = 77, [23, 33]

print("Before calling the function: ", a, b)
myfunc(a,b)
print("After calling the function: ", a, b)
```

77 , [23, 33]

77, [23,33,17]

2. The following questions are related to the Twitter scraping lab assignment. Answer the questions. (10 points) (10 points)

(1) Briefly describe the steps of gathering tweets containing some keywords and the metadata that you can obtain from them. (You don't have to give the exact function name.)

(2) Name the two common methods of word vectorization and explain the differences between the two methods with an example text.

코사인 유사도

3.. The following questions are related to the Data exploration lab assignment. (10 points)

(1) List some useful charts you can use for uni-variate and bi-variate analysis.

1 2 3
4 nan 5
null 6 7

(2) Describe some methods of cleaning the data (in DataFrames) used in the Lab.

df.fillna 로 처리함, or df.drop(columns ~~~ , inplace = True)