1. Show the execution result of the following code (1) ~ (4), (6) ~ (10) and the order of the list data (A,B,C,D) for the given boxplot graph (5). (40 points)

(1)
```
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[0,0] = 0
print(df)
```

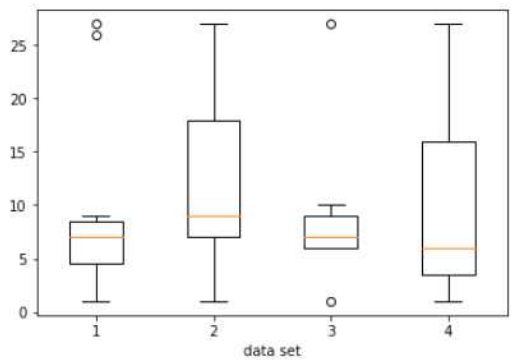
(2)
```
import numpy as np
data = np.array([[11,22,33],
                 [44,55,66],
                 [77,88,99]])
X, y = data[:, :-1], data[:, -1]
print(X); print(y)
```

(3)
```
from sklearn.feature_extraction.text import CountVectorizer
documents = ['This is the first document.',
             'Is this document the second document?']
cv = CountVectorizer(stop_words='english',
                     lowercase=True,
                     ngram_range=(1,2))
word_count = cv.fit_transform(documents)
print(cv.get_feature_names())
=> ['document', 'document second', 'second', 'second document']
print(word_count.toarray())
```

(4)
```
from bs4 import BeautifulSoup
html = """
   <html><body>
   <td>some text</td>
   <td><p>more text</p></td>
   </body></html>
"""
soup = BeautifulSoup(html, 'html.parser')
print(soup.td.text)
print(soup.p.text)
```

(5)
```
import matplotlib.pyplot as plt
 A = [1,2,3,4,5,6,6,7,25,26,27]
 B = [1,6,6,6,7,7,8,9, 9,10,27]
 C = [1,5,7,7,8,9,13,15,21,25,27]
 D = [1,4,4,5,5,7,7,8,9,26,27]

 box_plot_data=[ ?, ?, ?, ?]  # order ?
 plt.boxplot(box_plot_data)
 plt.xlabel('data set')
 plt.show()
 print(box_plot_data)
```



(6)
```
text = 'Write a Python code that accept some words and explore on this data'
with open("words.txt", "w") asf:
    f.write(text)
with open("words.txt", "r") asf:
    data = f.read()
print(len(data.split(" ")))
```

(7)
```
class myClass:
    def __init__(self, a: list):
        self.a = np.array(a)
    def get_result(self):
        return(np.argmax(self.a) * max(self.a))
class_1 = myClass([1, 4, 2, 3])
class_2 = myClass([5, 6, 9, 8])
a = class_1.get_result()
b = class_2.get_result()
print(a + b)
```

(8)
```
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()}))
```

(9)
```
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)
```

(10)
```
import numpy as np
N = 100000
ev0, ev1 = 0, 0
fori in range(N):
    if np.random.randint(0,2) == 0:
        ev0 +=1
    else:
        ev1 +=1
print(round(ev0/N,1), round(ev1/N,1))
```

2. We have discussed the following Python code in a class studying Web scraping. Describe how you modified (or changed) the code to do Lab assignment #2. You don't have to provide the exact Python code, but you must clearly show the modified program flow using Python syntax or pseudo code. (20 points)

```
url = 'https://kr.indeed.com/jobs?q=data+science&l=%EC%84%9C%EC%9A%B8%ED%8A%B9%EB%B3%
      84%EC%8B%9C'
link = requests.get(url)
soup = BeautifulSoup(link.text, 'html.parser')
job_elems = soup.select('.resultContent') # class

for i in job_elems:
    title = i.find('h2')
    company = i.find('span', class_='companyName')
    location = i.find('div', class_='companyLocation')

    if None in(title, company, location):
        continue

    print(title.text.strip())
    print(company.text.strip())
    print(location.text.strip())
```

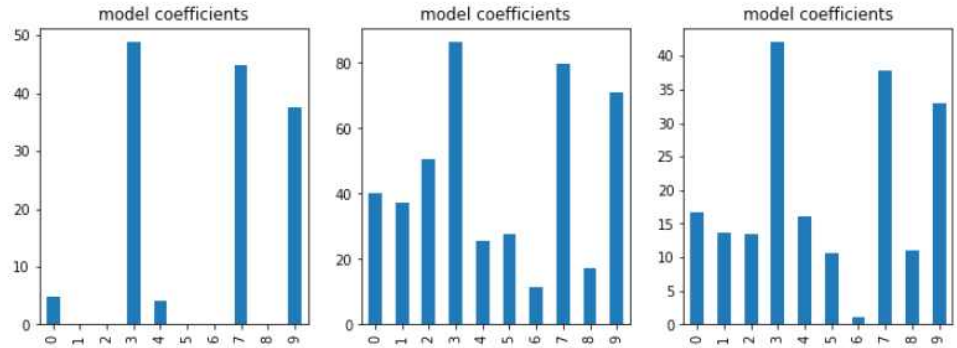
- -

3. Answer the questions by referring to the information given below (Python code, graphs). (20 points)

```
def show_plot(model, X, i):
    df = pd.DataFrame(X)
    coef = pd.Series(model.coef_, df.columns)
    plt.subplot(1,3,i)
    coef.plot(kind='bar', title='model coefficients')

X, y = make_regression(n_samples=100, n_features=10, noise=30, random_state=1)
X = StandardScaler().fit_transform(X)

clf1, clf2, clf3 = (____________, ____________, ____________)  # models
clfs = [clf1, clf2, clf3]
plt.figure(figsize=(12,4))
for i, clf in enumerate(clfs, 1):
    clf.fit(X, y)
    show_plot(clf, X, i)
```



(1) Explain what a **regularizer** is and briefly describe Ridge (L2) and Lasso (L1) regularizers using their respective (regression) loss functions.

(2) The graphs above shows the weights (or coefficients) for features as a result of training three linear regression models: LinearRegression( ), Ridge(alpha=100), and Lasso(alpha=30). Explain which one of them corresponds to each model along with your reasoning.

4. Explain concepts of the Gradient Descent algorithm using a graph of the loss function and a mathematical expression of the parameter updates. Use a model of a simple linear case with only one feature (x) and the target (y) as a training data set for both regression and classification. Assume the parameters of the model are w and b. (20 points)

	linear regression	logistic regression classifier
model diagram (with input, output, parameters)		
hypothesis: y_pred =		
loss function: Loss(w,b) =		
parameter update: w = b =		(hint) $\sigma(z) = 1/(1 + \exp(-z))$ $dLoss(w,b)/dz = \sigma(z) - y$