Machine Learning Report

01

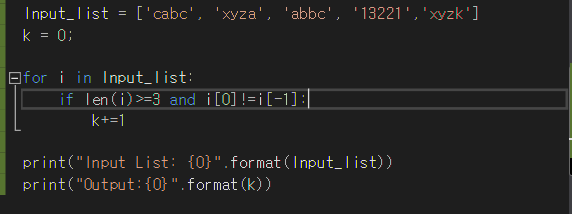
학번: 2017112200

이름: 신현호

1. Write a Python program to count the number of strings where the string length is 3 or more and the first and last character are different from a given list of strings.

Input List : ['cabc', 'xyza', 'abbc', '13221', 'xyzk'] Output : 3

<Program Code>



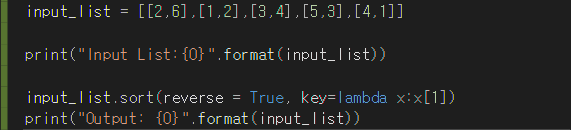
<Result>



1. Write a Python program to get a list, sorted in decreasing order by the second element in each inner list from a list, using a 'lambda' function.

Input List : [[2, 6], [1, 2], [3, 4], [5, 3], [4, 1]] Output : [[2, 6], [3, 4], [5, 3], [1, 2], [4, 1]]

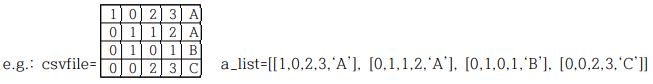
<Program Code>



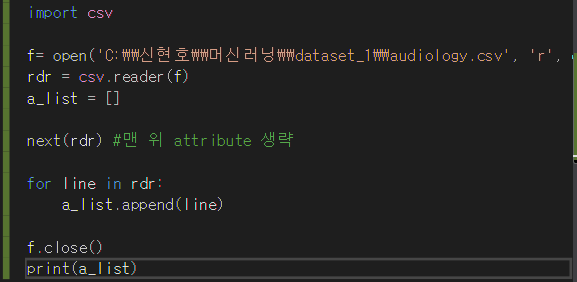
<Result>



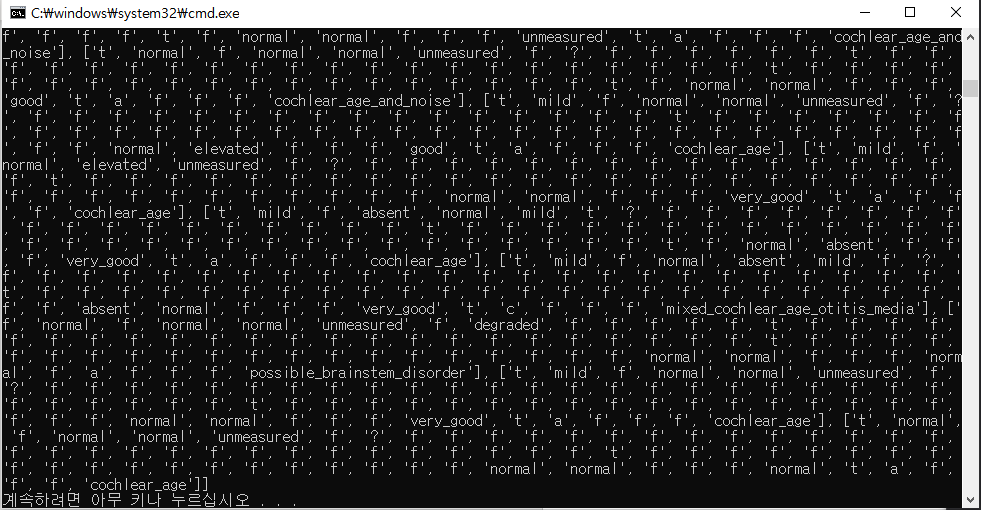
1. Write Python code for the following tasks
2. read csvfile into a two dimension list (called “a\_list”) using ‘csv’ module.



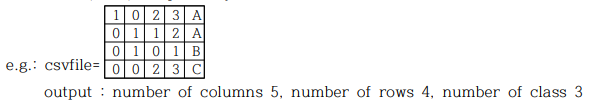
<Program Code>



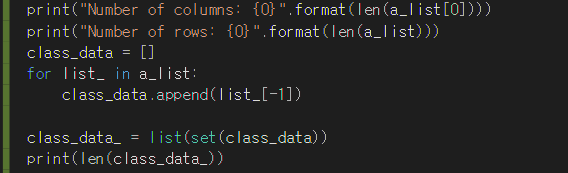
<Result>



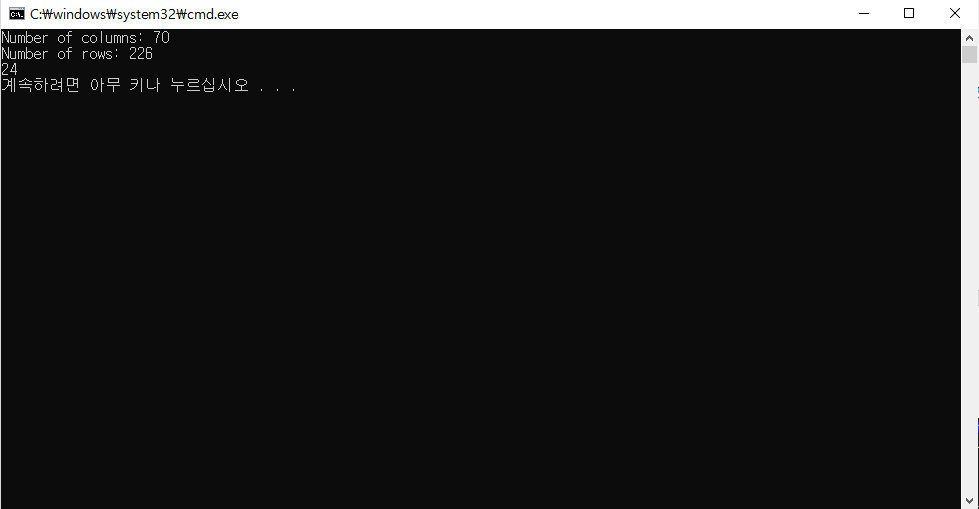
1. show the number of columns(attributes), number of rows(records) and number of classes (label), respectively.



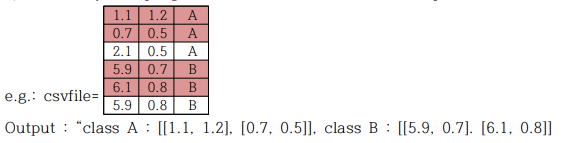
<Program Code>



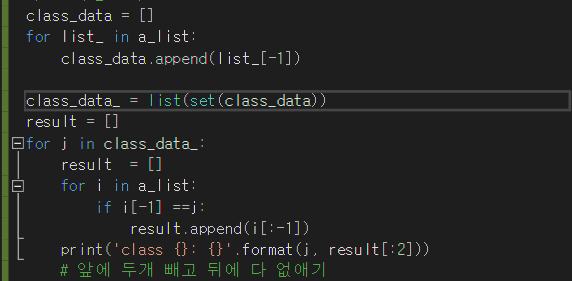
<Result>



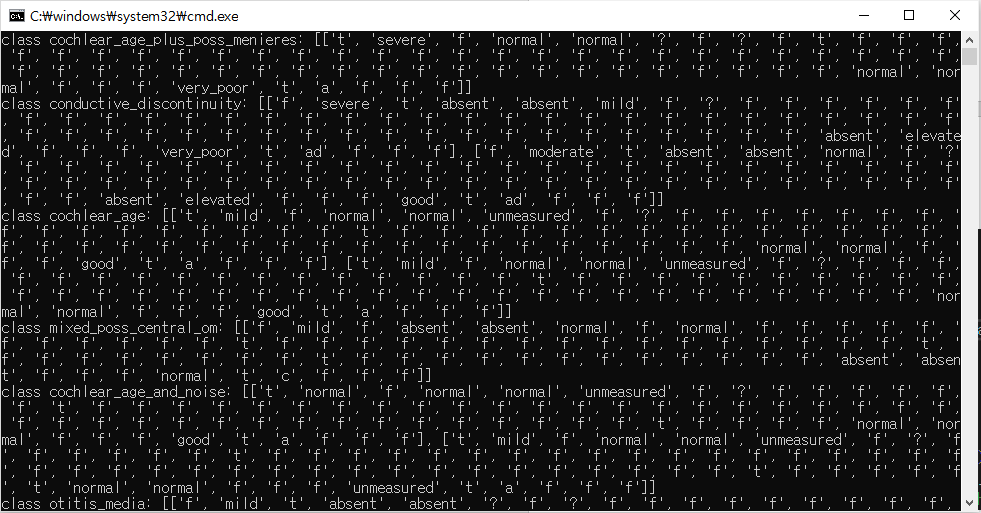
1. write a Python program that shows the first 2 rows per class from the “a\_list”



<Program Code>

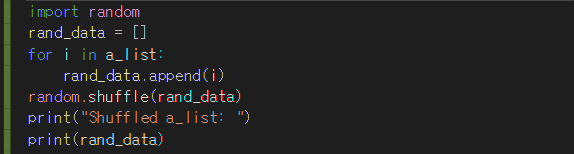


<Result>



1. write a Python program which randomly shuffles ‘a\_list’ data using 'random' module.

<Program Code>

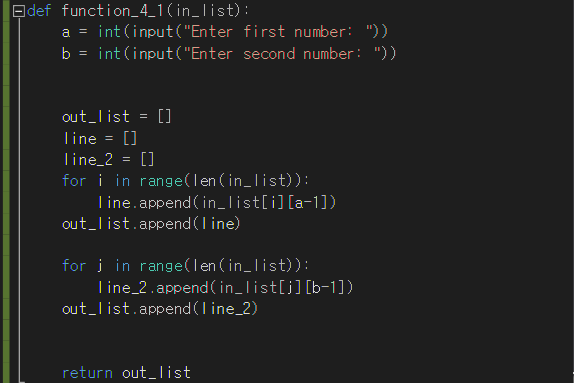


<Result>



1. Using the “a\_list” in question 3. write Python code for the following tasks
   1. given two column(attribute) index numbers, write a function that return a two-dimensional list that have the values of the columns

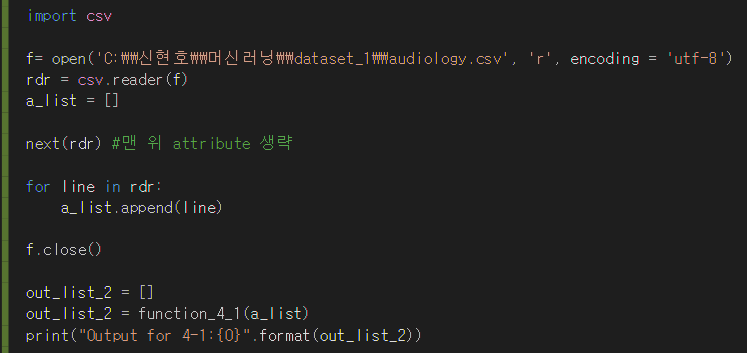
<Program code>



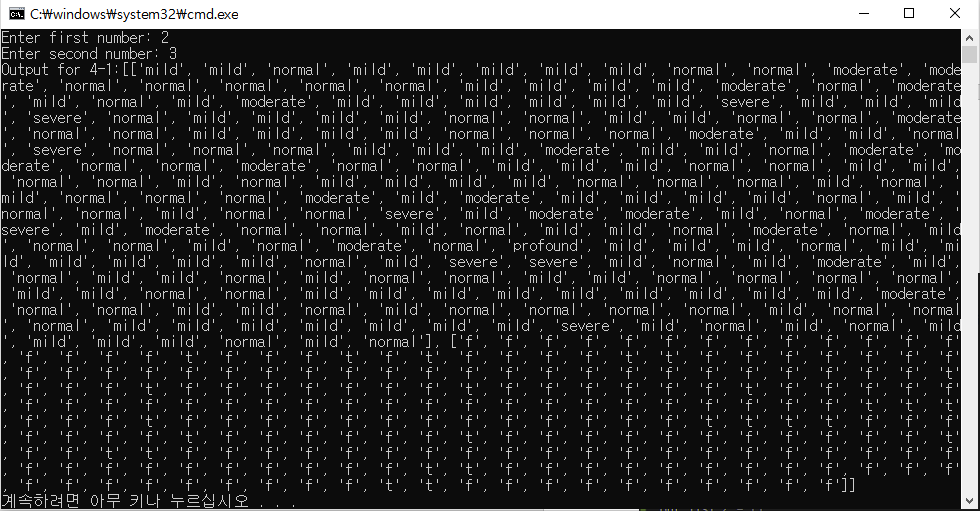
* 1. show the reversed elements of each list of q. 1) result. (We don’t actually change the values a\_list)

ex) q.1) output = [[0,1,2,0], [3,4,5,6]] q.2) output = [[0,2,1,0], [6,5,4,3]]

<Program Code>



<Result>

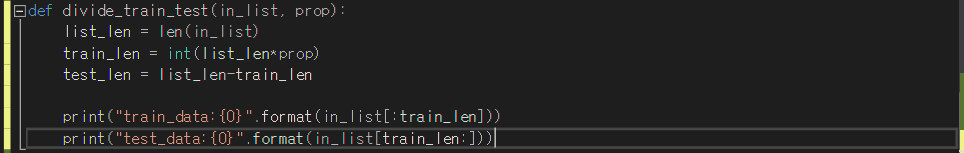


1. Using the “a\_list”, write Python code for the following tasks
   1. define a function “divide\_train\_test(in\_list, prop)” function where

input: 1) in\_list: a 2D list, 2) prop: proportion of training data

output: train\_data (first “prop” percent of in\_list), test\_data (the rest of in\_list)

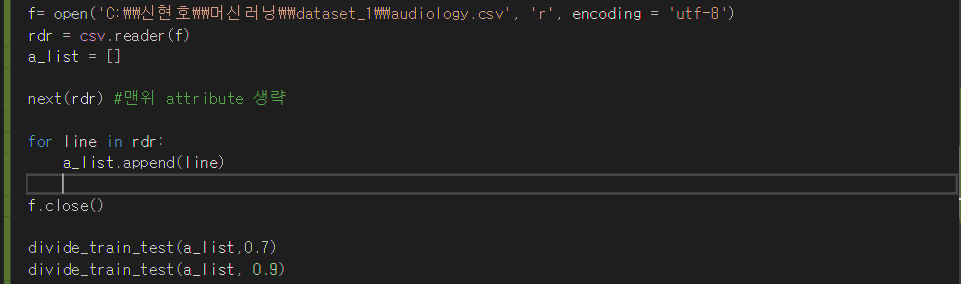
<Program Code>



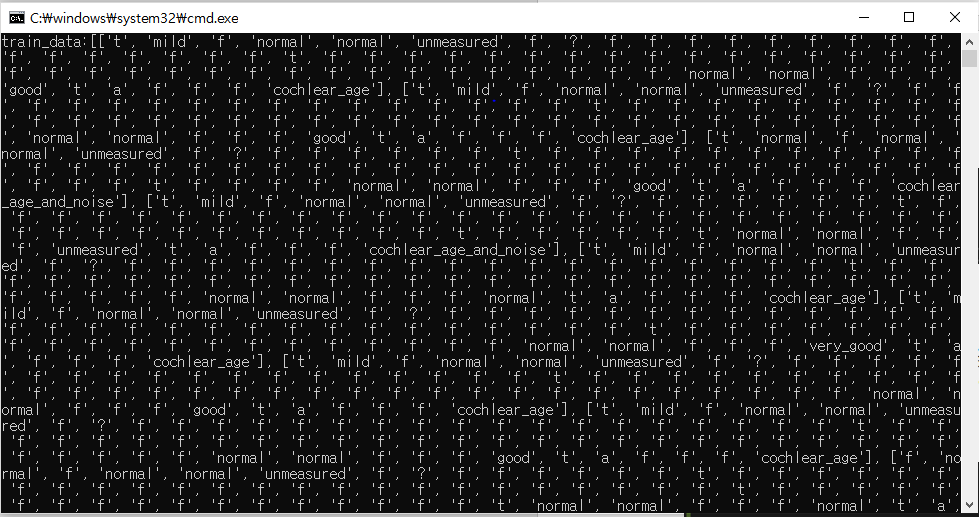
* 1. run divide\_train\_test(a\_list, prop) TWO times using prop=0.7, 0.9, respectively, and show the result.



<Program Code>



<Result>

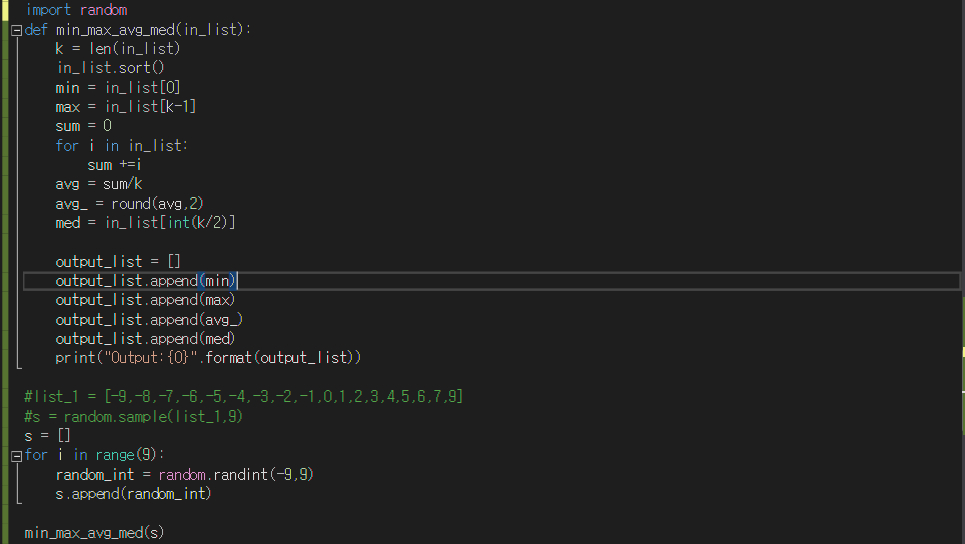


1. Write Python code for the tasks.
   1. define a function “min\_max\_avg\_med” which takes a list of numbers and returns [minimum, maximum, average, median] of the list. (don't use any modules such as 'import statistics') e.g.: def min\_max\_avg\_med(in\_list):
   2. Using ‘random’ module, randomly generate 9 integer numbers (the numbers are more than -10 and less than 10) and, calculate the minimum, maximum, average and median values of generated numbers using above “min\_max\_avg\_med” function.

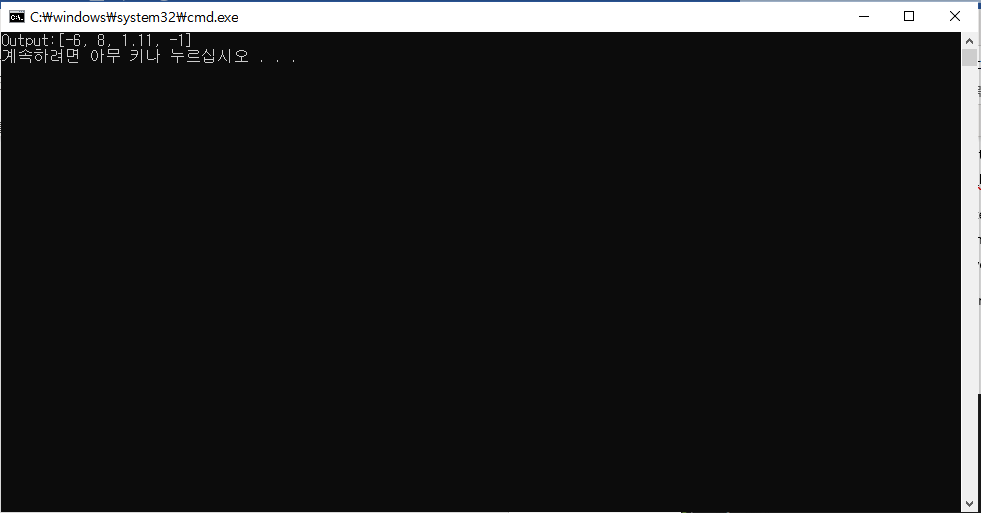
e.g.: min\_max\_avg\_med([-9,2,1,3,7,2,3,-7,-4]) returns [-9, 7, -0.222, 2]

<Program Code>

#6-1~6-2



<Result>



* 1. define a function “equ\_interval” which divides a value range into n equal intervals.

input: 1) list [min, max] of range, 2) number of intervals

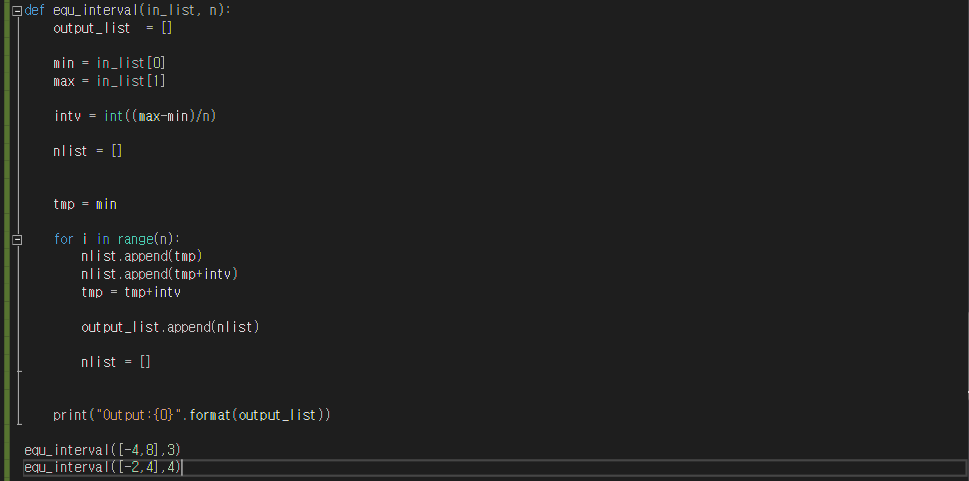
output: list of (equal distance) intervals

e.g.: equ\_interval([-4, 8], 3) returns [[-4,0], [0,4], [4,8]]

4) run equ\_interval 2 times by using different values of list and number of intervals.

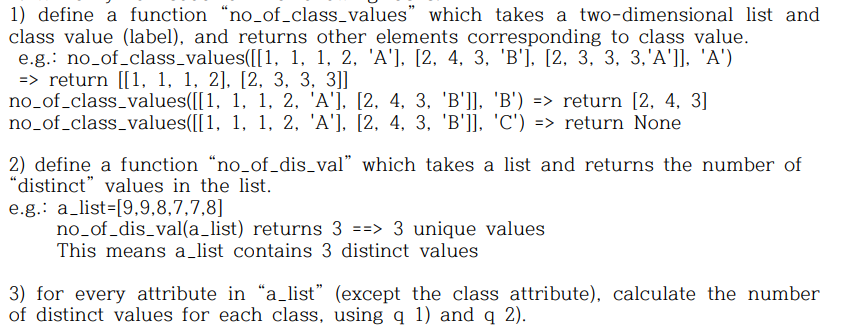
<Program Code>

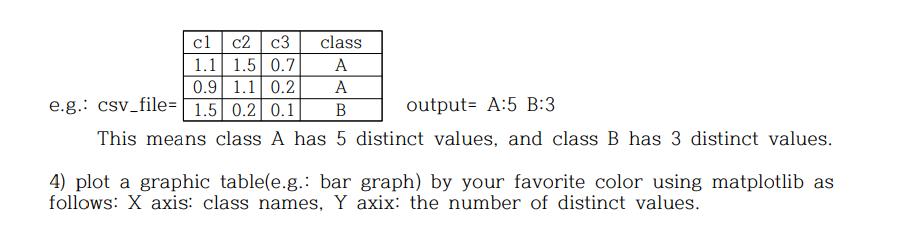
#6-3~6-4



<Result>

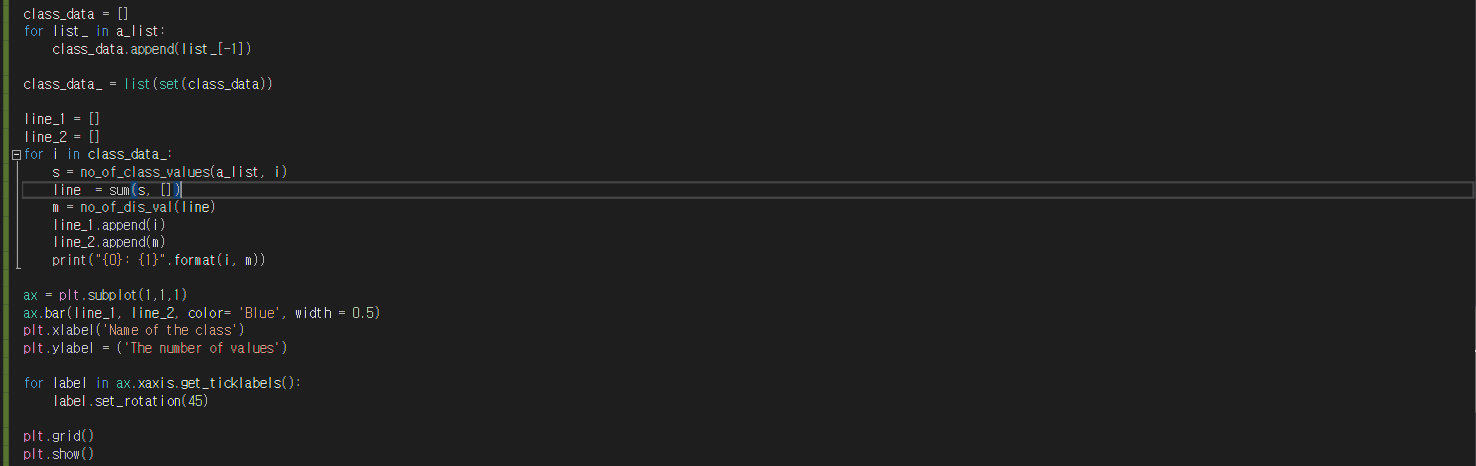






<Program Code>





<Result>

