Python Assignment 2: Data Structures - List, Dictionary, Set & Conditional Statements

List (Creation, Modification and Access):

1. List Creation:

a. Create a list named **age_list** with five integer elements. For eg., [24, 25, 26, 27, 28]

b. Create a list named **name_list** with five string elements.

2. List Operations / Modifications:

a. Append the string `"Yazhini"` to **name_list**.

b. Insert the element 30 at index 2 in **age_list**.

c. Remove the string `"Yazhini"` from **name_list**.

d. Pop the last element from **age_list**.

e. Extend the **age_list** with additional ages [29, 30, 26].

f. Sort **age_list** in descending order.

g. Find Max age, Min age and sum of all ages from **age_list**.

3. Accessing List Elements:

a. Print the first element of **name_list**.

b. Print the last element of **name_list**.

c. Print the elements from index 2 to index 4 in **name_list**.

d. Print the elements of **name_list** in reverse order.

Dictionary (Creation, Modification and Access):

a. Create a dictionary named `student_marks` that maps the names of five students to their marks (use scale of from 0 to 100).

b. Access and print the mark of a specific student, of your choice.

c. Add a new student "Janani" with a mark of 80 to the `student_marks` dictionary.

d. Update the mark of any one older student to 82.

e. Use the `keys()`, `values()`, and `items()` methods to print all keys, values, and key-value pairs in the `student_marks` dictionary.

Sets (Operations):

a. Create a set called **my_set** with following values:

```
['a','e','i','o','u','a','a','i']
```

Analyse the output and provide explanation for the same.

b. Attempt to change the value of **my_set**[4] = `'s'`. If code throws an error, provide an explanation.

c. Create two sets:

`set1` with values: `{1, 3, 5, 7, 9}`
`set2` with values: `{2, 3, 5, 8, 10}`

d. Compute and print the union and intersection of `set1` and `set2`.

Operators & Conditional Statements :

(IF, ELIF, ELSE)

Performance Category Program:

1. Prompt user for Input. Score range should be from 0 to 10 (both inclusive).

2. Find the performance category based on the input score using following criteria:
   a. **Above Average**: Score greater than 7
   b. **Average:** Score between 4 and 7(both inclusive)
   c. **Below Average**: Score lesser than 4

3. **Output**: Print the Performance category

4. **Additional Step:** You can give a prompt of your choice to each category.
For eg: If score below average “Need to Improve your performance, consistent practice will lead to better results”.

Sample Output:

```
Enter your score (0 to 10): 7
Average: Good effort! Keep practicing, there's room for improvement.
```