Jawaharlal Nehru Technological University Hyderabad
Kukatpally, Hyderabad - 500 085, Telangana , India

Name of Participant: Thummuru Sreenivasulu
Reg_no: 182023029
Course: AIML
Date: 12.02.2024

Assignment-2

Question:1

Task:-

Given the names and grades for each student in a class of N students, store them in a nested list and print the name(s) of any student(s) having the second lowest grade.

Note: If there are multiple students with the second lowest grade, order their names alphabetically and print each name on a new line.

Example

Records = [[“chi”, 20.0]], [“beta”, 50.0], [“alpha”, 50.0]

The ordered list of scores is [20.0, 50.0], so the second lowest score is 50.0. There are two students with that score: [“beta”, “alpha”]. Ordered alphabetically, the names are printed as:

```
alpha
beta
```

Input Format

The first line contains an integer , n, the number of students.

The 2N subsequent lines describe each student over lines.

- The first line contains a student’s name.
- The second line contains their grade.

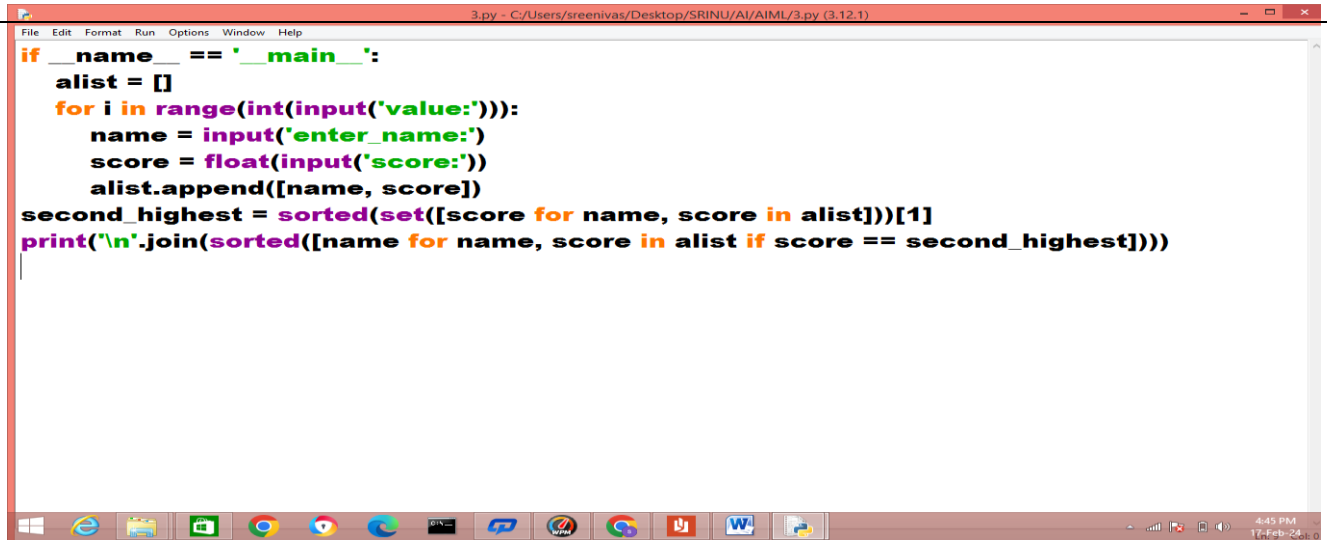
Constraints

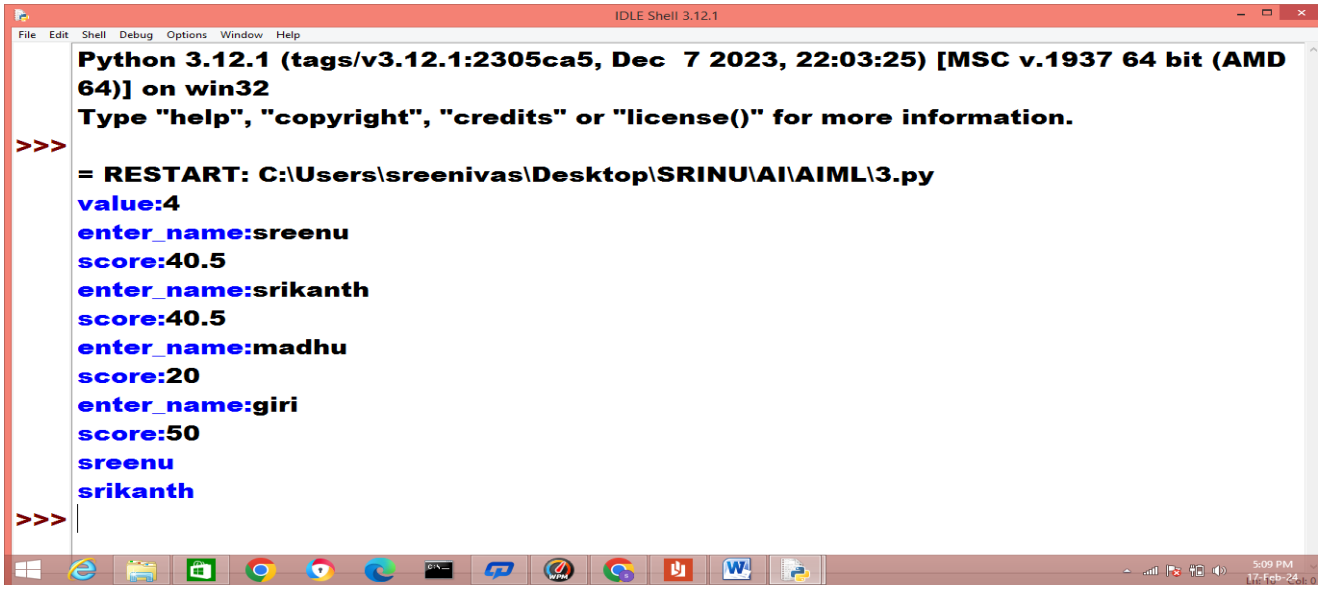
- 2 <= N <= 5
- There will always be one or more students having the second lowest grade.

Solution – Python

```
if __name__ == '__main__':
    alist = []
    for i in range(int(input('value:'))):
        name = input('enter_name:')
        score = float(input('score:'))
        alist.append([name, score])
second_highest = sorted(set([score for name, score in alist]))[1]
print('\n'.join(sorted([name for name, score in alist if score == second_highest])))
```

OUTPUT:





Explanation:

There are 4 students in this class whose names and grades are assembled to build the following list:

python students = [[‘sreenu’, 40.5], [‘srikanth’, 40.5],[madhu’, 20], [‘giri, 50]]

The lowest grade of 20 belongs to madhu. The second lowest grade of 40.5 belongs to both sreenu and srikanth, so we order their names alphabetically and print each name on a new line.

Question:2

Task:-

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
```

5

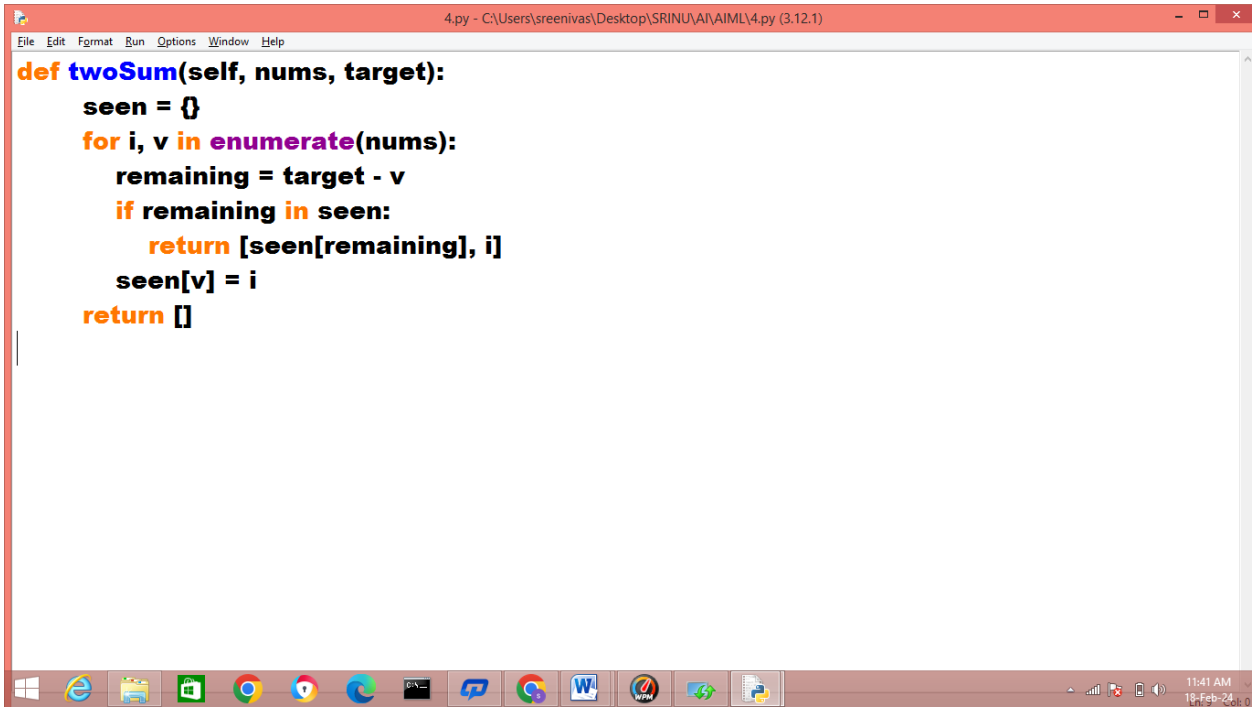
Solution – Python

```
def twoSum(self, nums, target):
    seen = {}
    for i, v in enumerate(nums):
        remaining = target - v
        if remaining in seen:
            return [seen[remaining], i]
        seen[v] = i
    return []
```

```
class Solution(object):
    def twoSum(self, nums, target):
        d = {}
```

```
        for i, num in enumerate(nums):
            t = target - num
            if t in d:
                return [d[t], i]
            d[num] = i
        return []
```

OUTPUT:



Input: nums = [2, 7, 11, 15], target = 9

Output: [0, 1]

Output: Because nums [0] + nums [1] == 9, we return [0, 1].

Constraints:

2 <= nums. Length <= 10^4

~~$-10^9 <= nums[i] <= 10^9$~~

$-10^9 <= target <= 10^9$

Only one valid answer exists.