Given the names and grades for each student in a class of N students, store them in a nested lsit and print the names of any students having the second lowest grade. Note: If there are multiple students with second lowest grade, order their names alphabetically and print each name on a new line

In [14]:
```
def second_lowest_grade(students):
    # Sort the students based on their grades
    students.sort(key=lambda x: x[1])

    # Find the second lowest grade
    second_lowest = None
    for student in students:
        if second_lowest is None or student[1] > second_lowest:
            second_lowest = student[1]
            break

    # Find students with the second lowest grade
    second_lowest_students = [student[0] for student in students if student[1] == second_lowest]

    # Sort the names alphabetically and print each name on a new line
    second_lowest_students.sort()
    for name in second_lowest_students:
        print(name)

# Test the function with some sample inputs
students = [
    ["Phani", 75],
    ["Sankara", 85],
    ["Satya", 75],
    ["Datta", 95],
    ["Chintalapati", 70]
]

print("Students with the second lowest grade:")
second_lowest_grade(students)
```

```
Students with the second lowest grade:
Chintalapati
```

Given an array of integers 'nums' and an integer targer, retun indices of the two numbers such that they add up to a 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

In [22]:
```
def two_sum(nums, target):
    num_dict = {}  # Dictionary to store complement of each number

    for i, num in enumerate(nums):
        complement = target - num

        # Check if complement exists in the dictionary
        if complement in num_dict:
            # Return the indices of the two numbers
            return [num_dict[complement], i]

        # Store the index of the current number
        num_dict[num] = i

    # If no solution found, return None
    return None

# Test the function with some sample inputs
nums = [2, 7, 11, 15, 4, 5, 2]
target = 9
print("Indices of the two numbers:", two_sum(nums, target))

nums = [3, 2, 4]
target = 6
print("Indices of the two numbers:", two_sum(nums, target))

nums = [3, 1, 5, 8 , 0]
target = 8
print("Indices of the two numbers:", two_sum(nums, target))
```

```
Indices of the two numbers: [0, 1]
Indices of the two numbers: [1, 2]
Indices of the two numbers: [0, 2]
```

In [ ]: