In [3]:
```
def find_second_lowest_grade(records):
    # Flatten the list
    flattened_records = [item for sublist in records for item in sublist]

    # Sort the records based on grades
    sorted_records = sorted(flattened_records, key=lambda x: (x[1], x[0]))

    # Find the second lowest grade
    second_lowest_grade = sorted(set(record[1] for record in sorted_records))[1]

    # Find students with the second lowest grade
    students_with_second_lowest = [record[0] for record in sorted_records if record[1]

    return students_with_second_lowest

# Test the function
if __name__ == "__main__":
    records = []
    while True:
        record_input = input("Enter student's name and grade (or 'done' to finish): ")
        if record_input.lower() == 'done':
            break
        else:
            # Split input to get name and grade and create the nested list
            name, grade = record_input.split(',')
            records.append([[name.strip(), float(grade.strip())]])

    second_lowest_students = find_second_lowest_grade(records)
    if second_lowest_students:
        print("Student(s) with the second lowest grade:")
        for student in sorted(second_lowest_students):
            print(student)
    else:
        print("There are no students with the second lowest grade.")
```

```
Enter student's name and grade (or 'done' to finish): chi,20.0
Enter student's name and grade (or 'done' to finish): beta,50.0
Enter student's name and grade (or 'done' to finish): alpha,50.0
Enter student's name and grade (or 'done' to finish): done
Student(s) with the second lowest grade:
alpha
beta
```

In [ ]:

In [5]:
```python
def two_sum(nums, target):
    num_index_map = {}

    for i, num in enumerate(nums):
        complement = target - num
        if complement in num_index_map:
            return [num_index_map[complement], i]
        num_index_map[num] = i

    return []

# Test the function with user input
if __name__ == "__main__":
    nums = list(map(int, input("Enter the array of integers separated by commas: ").sp
    target = int(input("Enter the target sum: "))
    result = two_sum(nums, target)
    if result:
        print("Indices of the two numbers that add up to the target:", result)
        explanation = "Explanation: The sum of {} at index {} and {} at index {} equal
            nums[result[0]], result[0], nums[result[1]], result[1], target)
        print(explanation)
    else:
        print("No two numbers found that add up to the target.")
```

```
Enter the array of integers separated by commas: 2,7,11,15
Enter the target sum: 9
Indices of the two numbers that add up to the target: [0, 1]
Explanation: The sum of 2 at index 0 and 7 at index 1 equals the target 9.
```

In [ ]:

In [ ]: