```python
records = [["chi", 20.0], ["beta", 50.0], ["alpha", 50.0]]

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

# Find the second lowest grade
second_lowest_grade = sorted_records[1][1] if len(sorted_records) > 1 else float('inf')

# Find the names corresponding to the second lowest grade
second_lowest_names = [record[0] for record in sorted_records if record[1] == second_lowest_grade]

# Sort the names alphabetically
second_lowest_names.sort()

# Print the names
for name in second_lowest_names:
    print(name)
```

STDIN

Output:

```
alpha
beta
```

```python
def find_lowest_grade_students(records):
    lowest_grade = min([record[1] for record in records])
    lowest_grade_students = [student[0] for student in records if student[1] == lowest_grade]

    lowest_grade_students.sort()

    return lowest_grade_students

records = [["John", 85.0], ["Emma", 92.0], ["Michael", 78.0], ["Olivia", 91.0], ["William", 78.0]]
lowest_grade_students = find_lowest_grade_students(records)

if len(lowest_grade_students) > 0:
    print("Students with the lowest grade:")
    for student in lowest_grade_students:
        print(student)
else:
    print("No students have the lowest grade.")
```

STDIN

Input for the program (Optional)

Output:

```
Students with the lowest grade:
Michael
William
```

```python
def twoSum(nums, target):
    complements = {}  # Create an empty hash map to store complements
    for i in range(len(nums)):
        complement = target - nums[i]  # Calculate the complement
        if complement in complements:  # If the complement exists in the hash map
            return [complements[complement], i]  # Return the indices of the pair
        else:
            complements[nums[i]] = i  # Store the complement and its index in the hash map
    return []  # Return an empty list if no solution is found

nums = [2, 7, 11, 15]
target = 9
print(twoSum(nums, target))
```

STDIN

Input for the program (Optional)

Output:

```
[0, 1]
```