

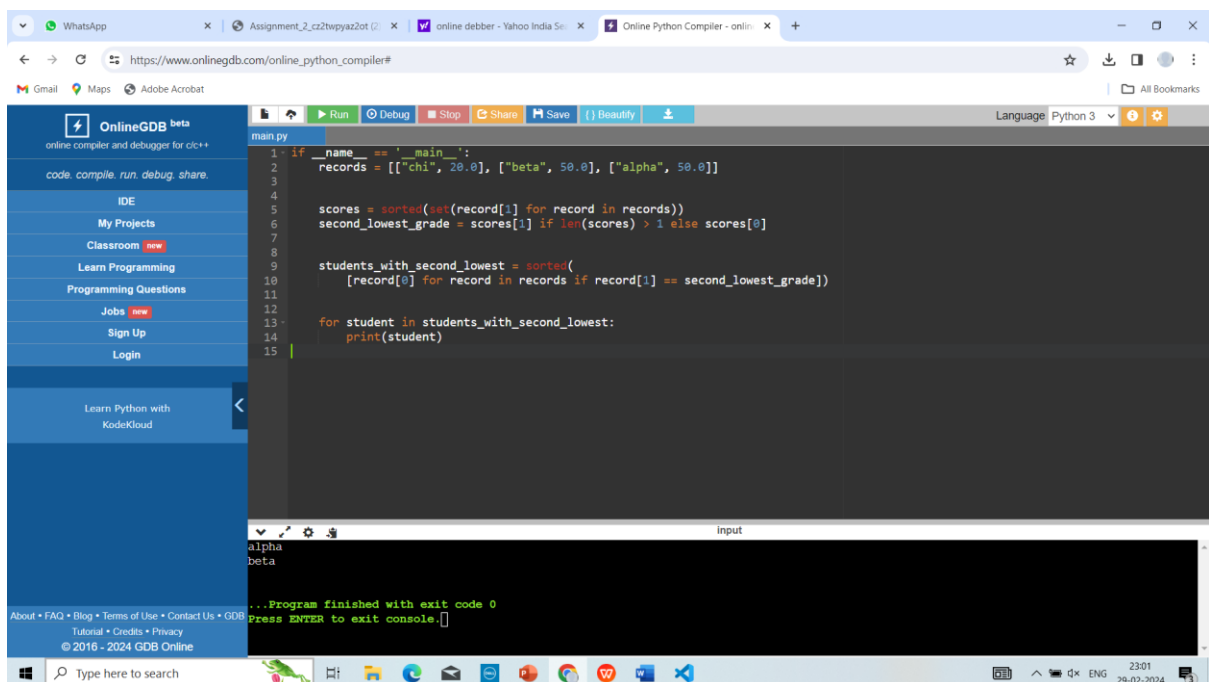
Code 1:

```
if __name__ == '__main__':  
    records = [{"chi", 20.0}, {"beta", 50.0}, {"alpha", 50.0}]  
  
    scores = sorted(set(record[1] for record in records))  
    second_lowest_grade = scores[1] if len(scores) > 1 else scores[0]  
  
    students_with_second_lowest = sorted(  
        [record[0] for record in records if record[1] == second_lowest_grade])  
  
    for student in students_with_second_lowest:  
        print(student)
```

OUTPUT:

alpha

beta



The screenshot shows a web browser window with the URL https://www.onlinegdb.com/online_python_compiler#. The browser tabs include WhatsApp, Assignment_2_c2twpyaz2ot, online debber - Yahoo India, and Online Python Compiler - online. The browser interface shows a search bar, navigation icons, and a sidebar with various links like 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Jobs', 'Sign Up', and 'Login'. The main content area displays a Python code editor with the following code:

```
1: if __name__ == '__main__':  
2:     records = [{"chi", 20.0}, {"beta", 50.0}, {"alpha", 50.0}]  
3:  
4:  
5:     scores = sorted(set(record[1] for record in records))  
6:     second_lowest_grade = scores[1] if len(scores) > 1 else scores[0]  
7:  
8:     students_with_second_lowest = sorted(  
9:         [record[0] for record in records if record[1] == second_lowest_grade])  
10:  
11:  
12:     for student in students_with_second_lowest:  
13:         print(student)  
14:  
15:
```

Below the code editor, there is an input field and an output console. The output console shows the following output:

```
alpha  
beta  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

The bottom of the screenshot shows the Windows taskbar with the search bar, task view icon, and several application icons. The system tray shows the date and time as 23:01 on 29-02-2024.

Code 2:

```
def two_sum(nums, target):
```

```
    # Create a dictionary to store the indices of elements
```

```
    num_dict = {}
```

```
    for i, num in enumerate(nums):
```

```
        complement = target - num
```

```
        # Check if the complement is already in the dictionary
```

```
        if complement in num_dict:
```

```
            # If found, return the indices of the two numbers
```

```
            return [num_dict[complement], i]
```

```
        # If not found, add the current number and its index to the dictionary
```

```
        num_dict[num] = i
```

```
    # If no solution is found, return an empty list (this should not happen in this problem)
```

```
    return []
```

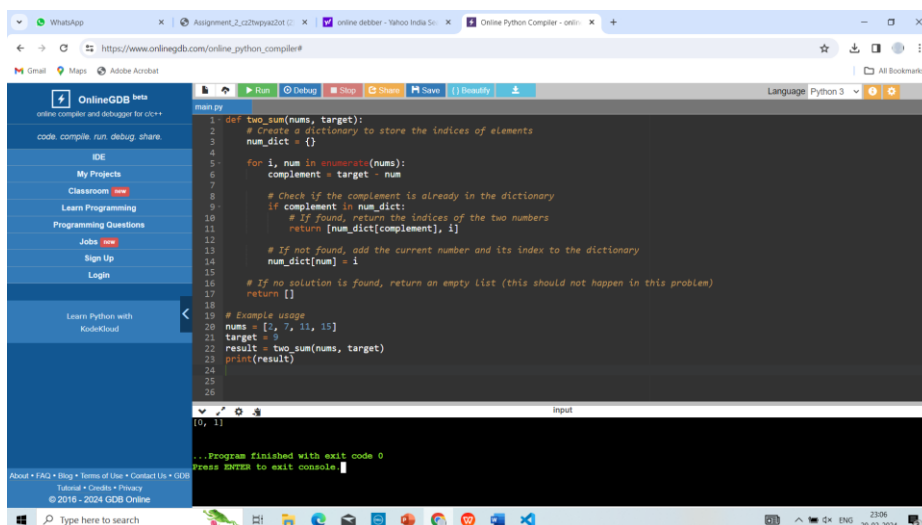
```
# Example usage
```

```
nums = [2, 7, 11, 15]
```

```
target = 9
```

```
result = two_sum(nums, target)
```

```
print(result) OUTPUT[0 1]
```



```
main.py
1: def two_sum(nums, target):
2:     # Create a dictionary to store the indices of elements
3:     num_dict = {}
4:
5:     for i, num in enumerate(nums):
6:         complement = target - num
7:
8:         # Check if the complement is already in the dictionary
9:         if complement in num_dict:
10:            # If found, return the indices of the two numbers
11:            return [num_dict[complement], i]
12:
13:         # If not found, add the current number and its index to the dictionary
14:         num_dict[num] = i
15:
16:     # If no solution is found, return an empty list (this should not happen in this problem)
17:     return []
18:
19: # Example usage
20: nums = [2, 7, 11, 15]
21: target = 9
22: result = two_sum(nums, target)
23: print(result)
24:
25:
26:
Input
[0, 1]
...Program finished with exit code 0
Press ENTER to exit console.
```

