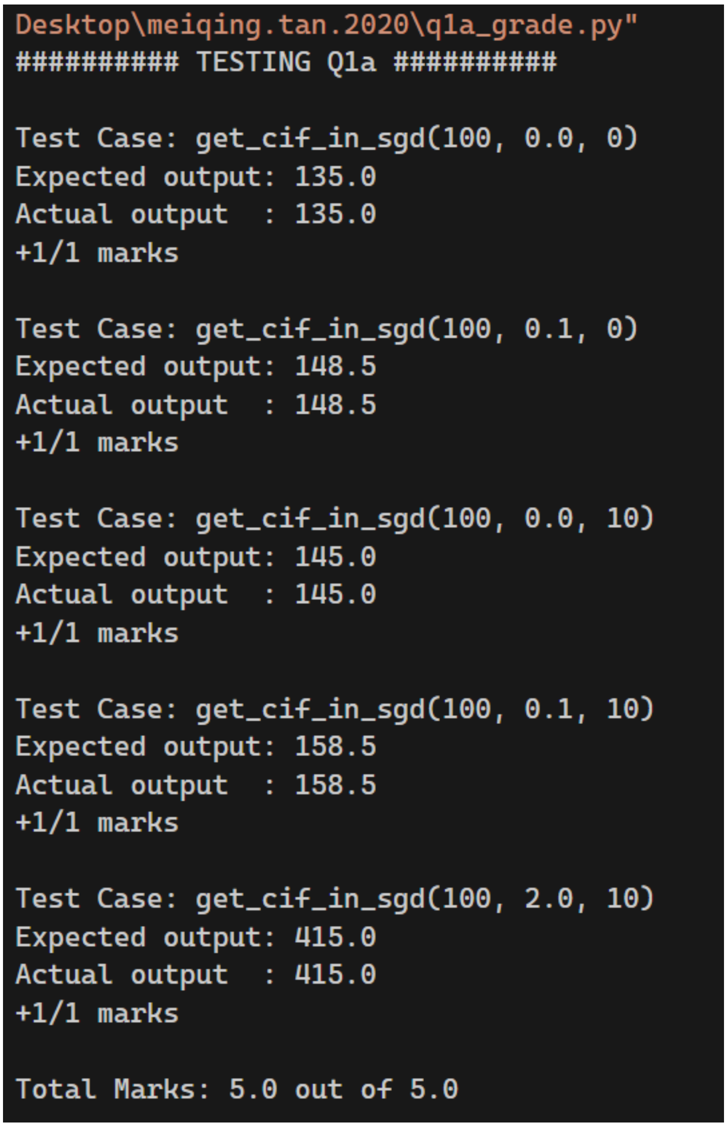
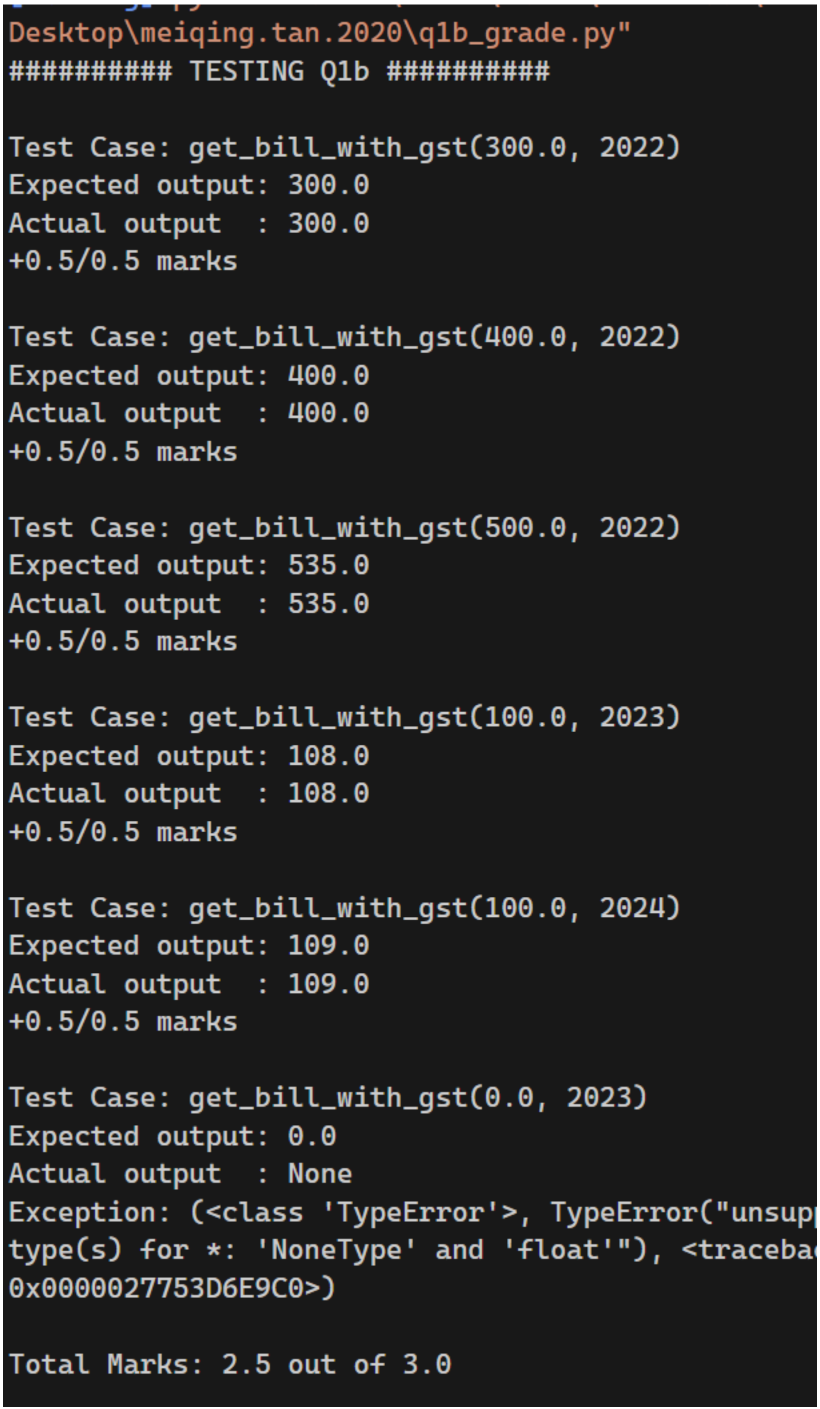
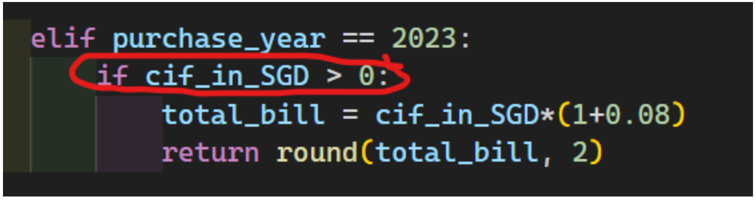


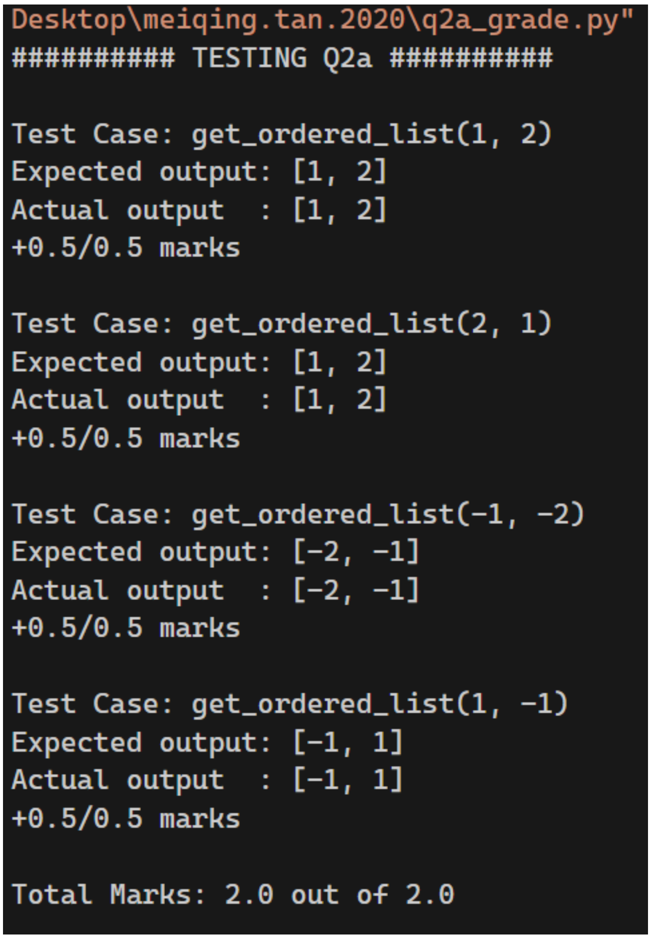
```
Desktop\meiqing.tan.2020\q1a_grade.py"
########## TESTING Q1a ##########

Test Case: get_cif_in_sgd(100, 0.0, 0)
Expected output: 135.0
Actual output  : 135.0
+1/1 marks

Test Case: get_cif_in_sgd(100, 0.1, 0)
Expected output: 148.5
Actual output  : 148.5
+1/1 marks

Test Case: get_cif_in_sgd(100, 0.0, 10)
Expected output: 145.0
Actual output  : 145.0
+1/1 marks

Test Case: get_cif_in_sgd(100, 0.1, 10)
Expected output: 158.5
Actual output  : 158.5
+1/1 marks

Test Case: get_cif_in_sgd(100, 2.0, 10)
Expected output: 415.0
Actual output  : 415.0
+1/1 marks

Total Marks: 5.0 out of 5.0
```



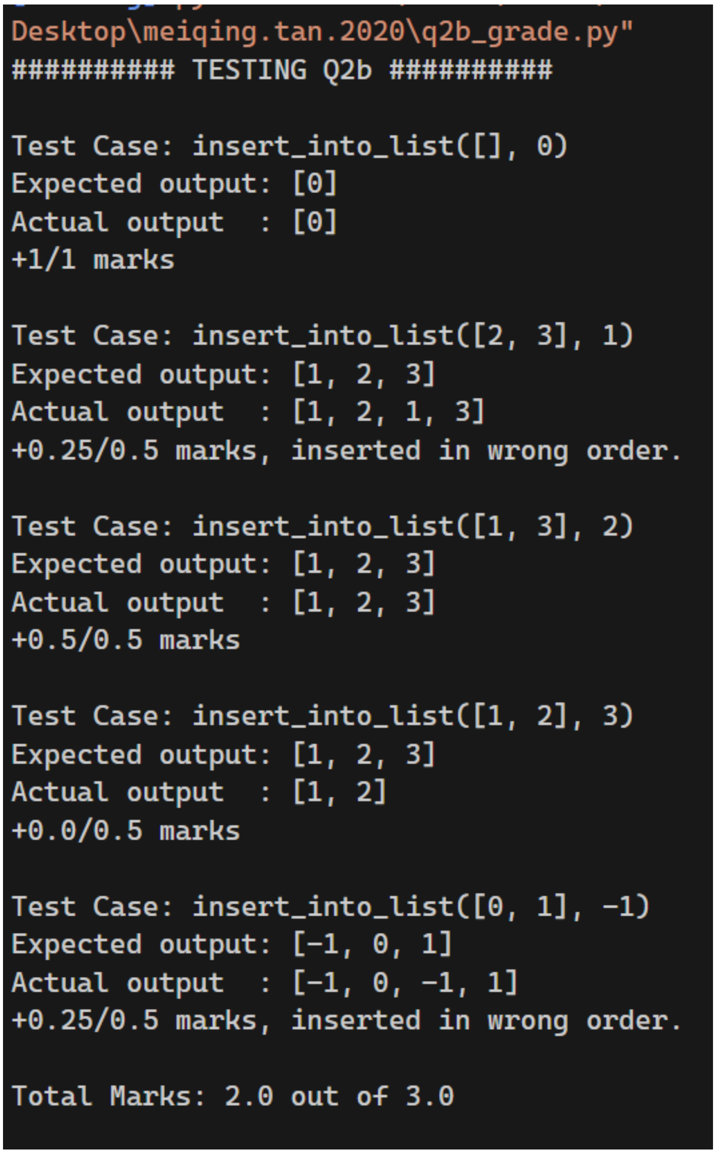
```
Desktop\meiqing.tan.2020\q1b_grade.py"
########## TESTING Q1b ##########

Test Case: get_bill_with_gst(300.0, 2022)
Expected output: 300.0
Actual output  : 300.0
+0.5/0.5 marks

Test Case: get_bill_with_gst(400.0, 2022)
Expected output: 400.0
Actual output  : 400.0
+0.5/0.5 marks

Test Case: get_bill_with_gst(500.0, 2022)
Expected output: 535.0
Actual output  : 535.0
+0.5/0.5 marks

Test Case: get_bill_with_gst(100.0, 2023)
Expected output: 108.0
Actual output  : 108.0
+0.5/0.5 marks

Test Case: get_bill_with_gst(100.0, 2024)
Expected output: 109.0
Actual output  : 109.0
+0.5/0.5 marks

Test Case: get_bill_with_gst(0.0, 2023)
Expected output: 0.0
Actual output  : None
Exception: (<class 'TypeError'>, TypeError("unsup
type(s) for *: 'NoneType' and 'float'"), <traceba
0x0000027753D6E9C0>)

Total Marks: 2.5 out of 3.0
```



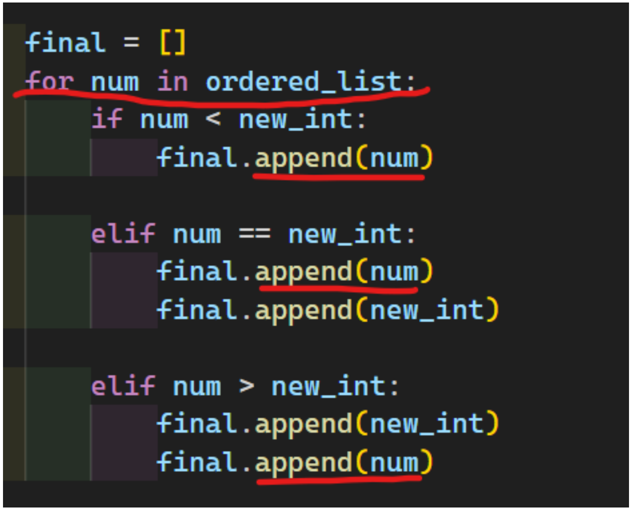
```
elif purchase_year == 2023:
    if cif_in_SGD > 0:
        total_bill = cif_in_SGD*(1+0.08)
        return round(total_bill, 2)
```



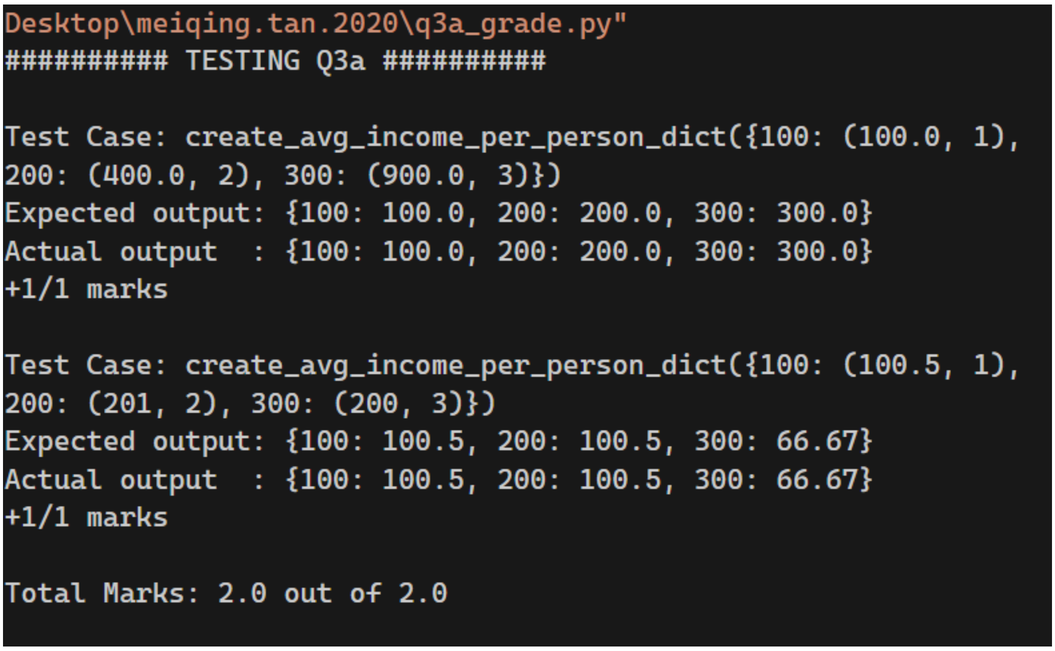
```
Desktop\meiqing.tan.2020\q2a_grade.py"
########## TESTING Q2a ##########

Test Case: get_ordered_list(1, 2)
Expected output: [1, 2]
Actual output  : [1, 2]
+0.5/0.5 marks

Test Case: get_ordered_list(2, 1)
Expected output: [1, 2]
Actual output  : [1, 2]
+0.5/0.5 marks

Test Case: get_ordered_list(-1, -2)
Expected output: [-2, -1]
Actual output  : [-2, -1]
+0.5/0.5 marks

Test Case: get_ordered_list(1, -1)
Expected output: [-1, 1]
Actual output  : [-1, 1]
+0.5/0.5 marks

Total Marks: 2.0 out of 2.0
```



```
Desktop\meiqing.tan.2020\q2b_grade.py"
########## TESTING Q2b ##########

Test Case: insert_into_list([], 0)
Expected output: [0]
Actual output  : [0]
+1/1 marks

Test Case: insert_into_list([2, 3], 1)
Expected output: [1, 2, 3]
Actual output  : [1, 2, 1, 3]
+0.25/0.5 marks, inserted in wrong order.

Test Case: insert_into_list([1, 3], 2)
Expected output: [1, 2, 3]
Actual output  : [1, 2, 3]
+0.5/0.5 marks

Test Case: insert_into_list([1, 2], 3)
Expected output: [1, 2, 3]
Actual output  : [1, 2]
+0.0/0.5 marks

Test Case: insert_into_list([0, 1], -1)
Expected output: [-1, 0, 1]
Actual output  : [-1, 0, -1, 1]
+0.25/0.5 marks, inserted in wrong order.

Total Marks: 2.0 out of 3.0
```



```python
final = []
for num in ordered_list:
    if num < new_int:
        final.append(num)

    elif num == new_int:
        final.append(num)
        final.append(new_int)

    elif num > new_int:
        final.append(new_int)
        final.append(num)
```



```
Desktop\meiqing.tan.2020\q3a_grade.py"
########## TESTING Q3a ##########

Test Case: create_avg_income_per_person_dict({100: (100.0, 1),
200: (400.0, 2), 300: (900.0, 3)})
Expected output: {100: 100.0, 200: 200.0, 300: 300.0}
Actual output  : {100: 100.0, 200: 200.0, 300: 300.0}
+1/1 marks

Test Case: create_avg_income_per_person_dict({100: (100.5, 1),
200: (201, 2), 300: (200, 3)})
Expected output: {100: 100.5, 200: 100.5, 300: 66.67}
Actual output  : {100: 100.5, 200: 100.5, 300: 66.67}
+1/1 marks

Total Marks: 2.0 out of 2.0
```