Name : Devansh koyani

Enr. No.: 22162171007

Batch: 54

Practical-1

(1) There are 2 chefs, namely chef 1 and chef 2 in the MasterChef competition. The judge is going to judge on the basis of 3 categories: presentation, taste and hygiene to prepare the dishes. The marking is scaling from 1 to 100. The rating for chef 1 challenge is the triplet a = (a[0], a[1], a[2]), and the rating for Chef 2 challenge is the triplet b = (b[0], b[1], b[2]), where 0 index is presentation, 1 index is taste and 2 index is hygiene.
The task is to find their comparison points by comparing a[0] with b[0], a[1] with b[1], and a[2] with b[2].

- If a[i] > b[i], then Chef 1 is awarded 1 point.
- If a[i] < b[i], then Chef 2 is awarded 1 point.
- If a[i] = b[i], then neither person receives a point.

Comparison points are the total points a person earned.
Given a and b, determine their respective comparison points.
Design the algorithm for the same and implement using the programming language of your choice. Make comparative analysis for various use cases & input size.

Sample Input 1

```
27 48 70
89 26 7
```

Sample Output 1

```
2 1
```

Explanation 1

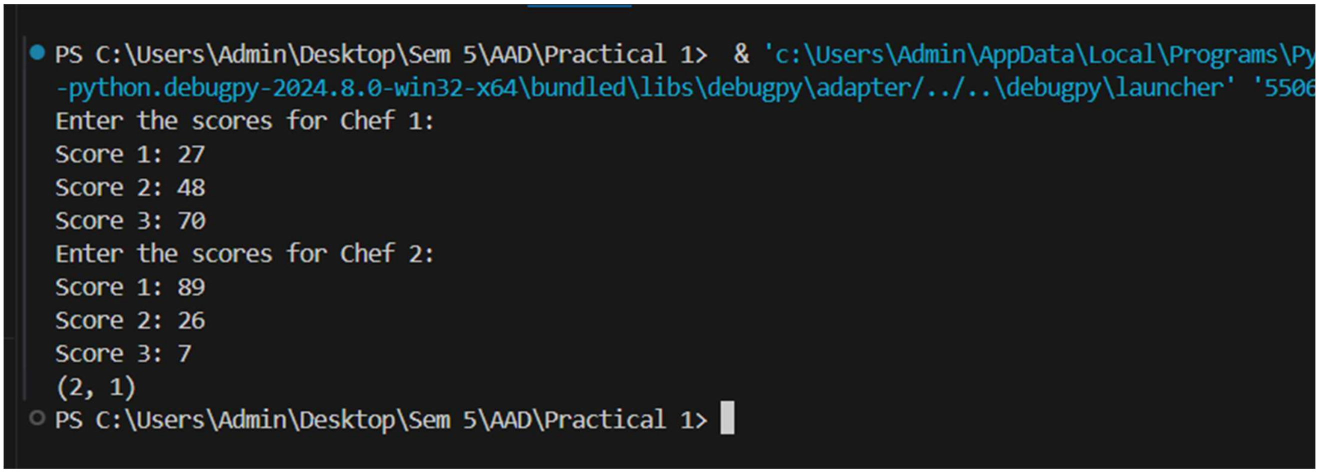
Comparing the 0th elements, 27<89 so Chef 2 receives a point.
Comparing the 1st and 2nd elements, 48>26 and 70>7 so Chef 1 receives two points. The return array is [2,1].

Code:

```
def compare(a, b):
    chef1 = 0
    chef2 = 0

    for i in range(3):
        if a[i] > b[i]:
            chef1 += 1
        elif a[i] < b[i]:
            chef2 += 1

    return chef1, chef2
```

```
a = []
b = []

print("Enter the scores for Chef 1:")
for i in range(3):
    score = int(input(f"Score {i+1}: "))
    a.append(score)

print("Enter the scores for Chef 2:")
for i in range(3):
    score = int(input(f"Score {i+1}: "))
    b.append(score)

result = compare(a, b)
print(result)
```

Output:



```
● PS C:\Users\Admin\Desktop\Sem 5\AAD\Practical 1>  & 'c:\Users\Admin\AppData\Local\Programs\Py
  -python.debugpy-2024.8.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '5506
  Enter the scores for Chef 1:
  Score 1: 27
  Score 2: 48
  Score 3: 70
  Enter the scores for Chef 2:
  Score 1: 89
  Score 2: 26
  Score 3: 7
  (2, 1)
○ PS C:\Users\Admin\Desktop\Sem 5\AAD\Practical 1> 
```

(2) Let us suppose that you are having an array containing both positive and negative numbers. Given the numbers you are supposed to find 2 such elements such that the sum of those numbers is closest to zero.

Sample Input 1

15, 5, -20, 30, -45

Sample Output 1

15, -20

Explanation 1

In all the comparison, the sum of 15 and -20 is smallest amount among all other comparison.

Sample Input 2

15, 5, -20, 30, 25

Sample Output 2

15, -20 & -20, 25

Explanation 2

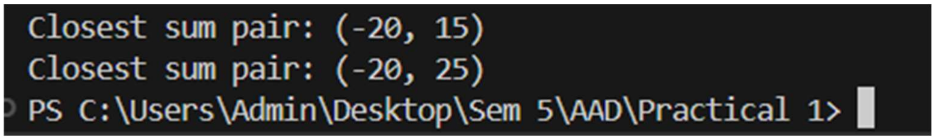
In all the comparison, the sum of 15,-20 & -20, 25 is smallest amount among all other comparison.

Code:

```
def find_closest_sum_pair(arr):
    arr.sort()
    left, right = 0, len(arr) - 1
    closest_pair = (arr[left], arr[right])
    min_sum = arr[left] + arr[right]

    while left < right:
        current_sum = arr[left] + arr[right]

        if abs(current_sum) < abs(min_sum):
            min_sum = current_sum
            closest_pair = (arr[left], arr[right])

        if current_sum < 0:
            left += 1
        else:
            right -= 1

    return closest_pair

arr1 = [15, 5, -20, 30, -45]
result1 = find_closest_sum_pair(arr1)
print(f"Closest sum pair: {result1}")

arr2 = [15, 5, -20, 30, 25]
result2 = find_closest_sum_pair(arr2)
```

```
print(f"Closest sum pair: {result2}")
```

Output:



```
Closest sum pair: (-20, 15)
Closest sum pair: (-20, 25)
PS C:\Users\Admin\Desktop\Sem 5\AAD\Practical 1>
```