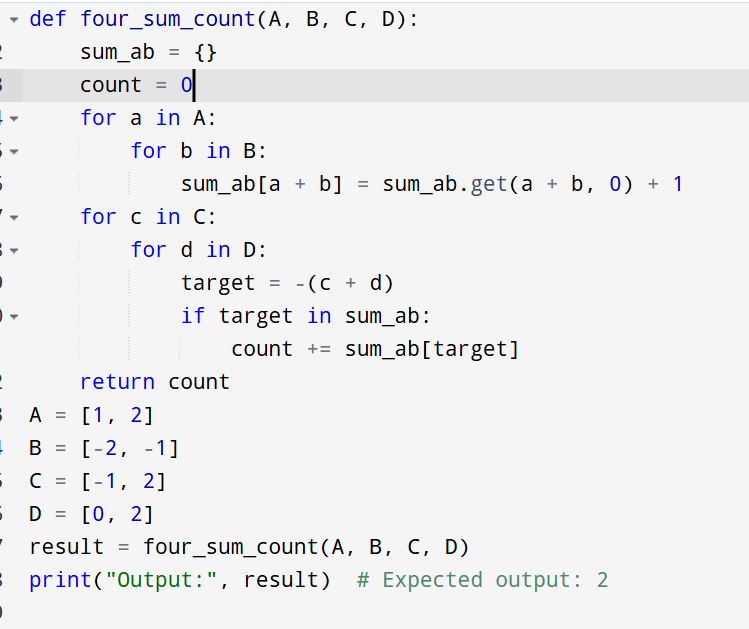
**3.10 TOTAL NUMBER OF TUPLES**

**Aim:** To find the total number of tuples (i,j,k,l)(i, j, k, l)(i,j,k,l) such that the sum of elements from four integer lists A[i]+B[j]+C[k]+D[l]=0A[i] + B[j] + C[k] + D[l] = 0A[i]+B[j]+C[k]+D[l]=0.

**Algorithm:**

1. Initialize a dictionary sum\_ab to store the frequency of sums of elements from lists A and B.
2. Calculate all possible sums of pairs from A and B:
   * For each element a in A:
     + For each element b in B:
       - Compute sum = a + b
       - Increment the count of this sum in sum\_ab.
3. Initialize a counter count to zero for storing the number of valid tuples.
4. Iterate over all pairs from C and D:
   * For each element c in C:
     + For each element d in D:
       - Calculate target = -(c + d)
       - If target exists in sum\_ab, add sum\_ab[target] to count.
5. Return count, which is the total number of tuples satisfying A[i]+B[j]+C[k]+D[l]=0A[i] + B[j] + C[k] + D[l] = 0A[i]+B[j]+C[k]+D[l]=0.

**Program:**

****

**Input:**

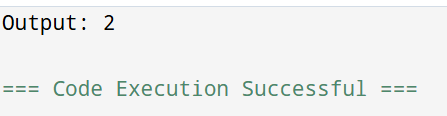
A = [1, 2]

B = [-2, -1]

C = [-1, 2]

D = [0, 2]

**Output:**

****

**Result:** Thus, the program is executed successfully and the output is verified.

**Performance analysis:**

| **Approach** | **Time Complexity** | **Space Complexity** |
| --- | --- | --- |
| Naive (4 loops) | O(n4)O(n^4) | O(1)O(1) |
| Optimized (HashMap) | O(n2)O(n^2) (for sums) | O(n2)O(n^2) for hashmap |