Question :

* Given five array , such that Find the number of tuplets (i,j,k,l,m) such that  -
* a[i] + b[j] + c[k] + d[l] + e[m] = 0

Observation :

* Change the eqn , a[i] + b[j] + c[k] = -(d[l] + e[m])

Step 1 : create an hashMap and an integer count

Step 2 : store the sum of all possibilities of sum (a[i] + b[j] + c[k]) which takes the time complexcity of O (N ^ 3).

Step 3 : Count the pairs , which has the sum , (-(d[l] + e[m]).

class Solution {

   private int countTuplets( int [] A , int [] B , int [] C , int [] D , int [] E){

    Map<Integer , Integer> map =new HashMap<>();

    int count = 0;

    for(int i = 0 ; i < A.length ; i++){

        for(int j = 0 ; j < B.length ; j++){

            for(int k = 0 ; k < C.length ; k++){

                int sum = A[i] + B[i] + C[i];

                map.put(sum , map.getOrDefault(sum , 0) + 1);

            }

        }

    }

    for(int l = 0 ; l < D.length ; l++){

        for(int m = 0 ; m < E.length ; m++){

            int sum = D[l] + E[m];

            count += map.getOrDefault(-sum , 0);

        }

    }

    return count;

   }

}