

# Find All Four Sum Numbers

---

Given an array of integers and another number. Find all the **unique** quadruple from the given array that sums up to the given number.

## Example 1:

**Input:** N = 5, K = 3

A[] = {0,0,2,1,1}

**Output:** 0 0 1 2 \$ **Explanation:** Sum of 0, 0, 1, 2 is equal to K.

## Example 2:

**Input:** N = 7, K = 23

A[] = {10,2,3,4,5,7,8}

**Output:** 2 3 8 10 \$ 2 4 7 10 \$ 3 5 7 8 \$ **Explanation:** Sum of 2, 3, 8, 10 = 23, sum of 2, 4, 7, 10 = 23 and sum of 3, 5, 7, 8 = 23.

```
class Solution:
    def fourSum(self, a, k):
        n=len(a)
        ans=[]
        if(n < 4):
            return ans
        a.sort()
        for i in range(0, n-3):
            # current element is greater than k then no quadruplet can be
            foundf
            if (a[i] > 0 and a[i] > k):
                break
            # removing duplicates
            if (i > 0 and a[i] == a[i - 1]):
                continue

            for j in range(i+1, n-2):
                # removing duplicates
                if (j > i + 1 and a[j] == a[j - 1]):
                    continue

            # taking two pointers
```

```
    left = j + 1
    right = n - 1
    while (left < right):
        old_l = left;
        old_r = right;
        # calculate current sum
        sum = a[i] + a[j] + a[left] + a[right]
        if (sum == k):
            # add to answer
            ans.append([a[i], a[j], a[left], a[right]])

            # removing duplicates
            while (left < right and a[left] == a[old_l]):
                left+=1
            while (left < right and a[right] == a[old_r]):
                right-=1
        elif (sum > k):
            right-=1
        else:
            left+=1

    return ans
```