**Find All Four Sum Numbers: -**

**Medium** Accuracy: **19.94%** Submissions: **172K+** Points: **4**

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

Also note that all the quadruples which you return should be internally sorted, ie for any quadruple [q1, q2, q3, q4] the following should follow: q1 <= q2 <= q3 <= q4.

**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.

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **fourSum()** which takes the array arr[] and the integer k as its input and returns an array containing all the quadruples in a lexicographical manner. In the output each quadruple is separate by $. The printing is done by the driver's code.

**Expected Time Complexity:** O(N3).  
**Expected Auxiliary Space:** O(N2).

**Constraints:**  
1 <= N <= 100  
-1000 <= K <= 1000  
-100 <= A[] <= 100

**Code: -**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

// User function template for C++

class Solution{

public:

// arr[] : int input array of integers

// k : the quadruple sum required

vector<vector<int> > fourSum(vector<int> &arr, int K) {

// Your code goes here

int n = arr.size();

vector<vector<int>> ans;

if(n < 4) return ans;

sort(arr.begin(), arr.end());

long long sum;

for(int i = 0; i<n-3; ++i){

// except for 1st case if "i" found to be duplicate

if(i > 0 and arr[i] == arr[i-1])

continue;

for(int j = i+1; j<n-2; ++j){

// except for the 1st case w.r.t "i" if "j" found to be duplicate

if(j > i+1 and arr[j] == arr[j-1])

continue;

int k = (j + 1), l = (n - 1);

while(k < l){

sum = arr[i];

sum += arr[j];

sum += arr[k];

sum += arr[l];

if(sum == K){

ans.push\_back({arr[i], arr[j], arr[k], arr[l]});

++k; --l;

while(k < l and arr[k] == arr[k-1]) ++k;

while(k < l and arr[l] == arr[l+1]) --l;

}

else if(sum < K) ++k;

else --l;

}

}

}

return ans;

}

};

//{ Driver Code Starts.

int main() {

int t;

cin >> t;

while (t--) {

int n, k, i;

cin >> n >> k;

vector<int> a(n);

for (i = 0; i < n; i++) {

cin >> a[i];

}

Solution ob;

vector<vector<int> > ans = ob.fourSum(a, k);

for (auto &v : ans) {

for (int &u : v) {

cout << u << " ";

}

cout << "$";

}

if (ans.empty()) {

cout << -1;

}

cout << "\n";

}

return 0;

}

// } Driver Code Ends

**T.C: - O(N3)**

**S.C: - O(1) excluding answer storage**