```
#include<iostream>
#include<vector>
#include<unordered_map>
#include<algorithm>
using namespace std;

//check whether pair exist in an array whose sum = k
int main(){
    vector<int> v = {1,2,4,6,7,9};
    int k = 8;
    bool present = false;
    unordered_map<int,int> umap;
    for(auto ele : v){
        umap[ele]++;
    }
    for(auto ele:umap){
        int key = ele.first;
        int value = ele.second;
        int pair = k-key;
        if(umap.find(pair)!=umap.end()){
            if(pair==key){
                if(value>1){
                    present = true;
                    break;
                }
            }
            else{
                if(umap.find(pair)!=umap.end()){
                    present = true;
                    break;
                }
            }
        }
    }
    if(present){
        cout<<"yes";
    }
    else{
        cout<<"no";
    }
    return 0;
}



//sort the array ele in pairs of +ve and -ve
// {11,-4,4,-3,-9,4,-4,-2,1,2,-4,2} => {-2,2,-4,4,-4,4}
int main(){
    vector<int> v = {11,-4,4,-3,-9,4,-4,-2,1,2,-4,2};
    unordered_map<int,int> umap;
    vector<int> negativeEle;
    vector<int> ans;
    for(auto ele:v){
```

```cpp
        if(ele>0){
            umap[ele]++;
        }
        else{
            negativeEle.push_back(ele);
        }
    }
    sort(negativeEle.begin(),negativeEle.end());
    for(int i=negativeEle.size()-1; i>=0; i--){
        int absValue = abs(negativeEle[i]);
        if(umap.find(absValue)!=umap.end()){
            if(umap[absValue]>0){
                umap[absValue]--;
                ans.push_back(negativeEle[i]);
                ans.push_back(absValue);
            }
        }
    }
    for(auto ele: ans){
        cout<<ele<<" ";
    }
    return 0;
}



//isomorphic string
bool isOneToOneMapping(string s1, string s2){
    unordered_map<char,char> umap;
    for(int i=0; i<s1.size(); i++){
        if(umap.find(s1[i])!=umap.end()){
            if(umap[s1[i]]!=s2[i]){
                return false;
            }
        }
        else{
            umap[s1[i]]=s2[i];
        }
    }
    return true;
}

bool isIsomorphic(string s1, string s2){
    if(s1.size()!=s2.size()){
        return false;
    }
    bool res1 = isOneToOneMapping(s1,s2);
    bool res2 = isOneToOneMapping(s2,s1);
    return res1 && res2;
}

int main(){
    string str1="abcdeag",str2="hijklmn";
    isIsomorphic(str1,str2) ? cout<<"yes" : cout<<"no";
```

```
        return 0;
    }
```