C++ Assignment Solutions | Hash Maps | Week 19

1. Relative Sort Array [LC 1122]

   Sol:

```
class Solution {
public:
    vector<int> relativeSortArray(vector<int>& arr1, vector<int>& arr2) {
        map<int,int> mp;
        for(auto &i : arr1) {
            mp[i]++;
        }
        vector<int> ans;
        for(auto &i : arr2) {
            while(mp[i]--)
                ans.push_back(i);
        }
        for(auto [i,j] : mp) {
            while(j-- > 0)
                ans.push_back(i);
        }
        return ans;
    }
};
```

2. Max number of k-sum pairs [LC 1679]

```
class Solution {
public:
    int maxOperations(vector<int>& nums, int k) {
        unordered_map<int,int> mp;
        int ans = 0;
        for(auto &i : nums) {
            if(mp[k - i] > 0) {
                mp[k - i]--;
                ans++;
            } else {
                mp[i]++;
            }
        }
        return ans;
    }
};
```

3. Finding Pairs with a Certain Sum [LC1865].

```
class FindSumPairs {
public:
    vector<int>v1,v2;
    map<int,int>mp;
    FindSumPairs(vector<int>& nums1, vector<int>& nums2) {
        v1 = nums1;
        v2 = nums2;
        for(auto val: v2)
            mp[val]++;
    }
```

```
void add(int index, int val) {
        if(index>=v2.size()){
            mp[val]++;
            return v2.push_back(val);
        }
        mp[v2[index]]--;
        v2[index]+=val;
        mp[v2[index]]++;
    }
    int count(int tot) {
        int cnt=0;
        for(auto val: v1){
            if(mp.find(tot-val)!=mp.end())
                cnt+=mp[tot-val];
        }
        return cnt;
    }
};
```

4. Bottom view of binary tree

```
void printBottomView(Node * root){
    if(!root) return ;
    unordered_map<int,int> mp;
    int l = 0;
    queue<pair<Node*,int>> q;
    q.push({root,0});
    while(!q.empty()){
        auto top = q.front();
        q.pop();
        Node *f = top.first;
        int ind = top.second;
        mp[ind] = f->data;
        l = min(ind,l);
        if(f->left){ q.push({f->left,ind-1});}
        if(f->right){ q.push({f->right,ind+1});}
    }
    while(mp.find(l)!=mp.end()){ cout<<mp[l++]<<" "; }
}
```

Note:- Please try to invest time doing the assignments which are necessary to build a strong foundation. Do not directly Copy Paste using Google or ChatGPT. Please use your brain😀.
