C++ Assignment Solutions | Priority Queues 2 | Week 20

1. Reorganize string LEETCODE:-767

   Sol:

```
string reorganizeString(string s) {
        priority_queue<pair<int, char>> pq;
        int n = s.size(); vector<int> cnt(26,0);
        for(auto i : s){
            cnt[i - 'a']++;
            if(2*cnt[i - 'a'] > n + 1) return "";
        }
        for(int i = 0; i < 26; i++)
            pq.push({cnt[i], i + 'a'});
        string ans(n,'?'); int curr = 0;
        while(pq.size()){
            auto temp = pq.top(); pq.pop();
            int p = temp.first;
            while(p--){
                ans[curr] = temp.second;
                curr += 2;
                if(curr >= n) curr = 1;
            }
        }
        return ans;
    }
```

2. Kth smallest prime fraction LEETCODE:-786

   Sol:

```
    vector<int> kthSmallestPrimeFraction(vector<int>& arr, int k){
        priority_queue<pair<double,pair<int,int>>>p;

        for(int i=0;i<arr.size();i++){
            for(int j=i+1;j<arr.size();j++){
                double a=(double)arr[i]/arr[j];
                p.push({a,{arr[i],arr[j]}});
                if(p.size()>k)
                p.pop();
            }
        }
        vector<int>ans;
        ans.push_back(p.top().second.first);
        ans.push_back(p.top().second.second);
        return ans;
    }
```

3. Minimum deviation in array LEETCODE:-1675
   Sol:

```
 int minimumDeviation(vector<int>& nums) {
        priority_queue<int> pq;

        int mn = 2e9, ans = mn;

        for(auto &i : nums) {
            if(i & 1) i *= 2;
            pq.push(i);
            mn = min(mn, i);
        }

        // now we have to minimize diff b/w min and max
        // 1st choice : decrease the max
        // we can do it by /= 2 if even and stop otherwise because max wont
change
        // 2nd choice : increase the min
        // can't do it as we have only even elements
        // if it was off then => It was even at some point(We started with all
evens)
        // => 2*(current min) > current_max
```

```
        ans = min(ans, (int)pq.top() - mn);

        while(pq.top() % 2 == 0) {
            int val = pq.top();
            pq.pop();
            val /= 2;
            pq.push(val);
            mn = min(mn, val);
            ans = min(ans, (int)pq.top() - mn);
        }

        return ans;
    }
```

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😀.
