Char Array and String - 03

Q.2325 Decode the Message :

LeetCode Solution:

```
class Solution {
public:
    void createMapping(unordered_map<char,char> &mapping, string key) {
        char start = 'a';

        for(int i=0; i<key.length(); i++) {
            //if mapping is not already present , then create it

            if(mapping.find(key[i]) == mapping.end()) {
                mapping[key[i]] = start;
                start++;
            }
        }

    }

    string useMapping(unordered_map<char,char> &mapping, string message) {
        string ans = "";
        for(int i=0; i<message.length(); i++) {
            char mappedCharacter = mapping[message[i]];
            ans.push_back(mappedCharacter);
        }
        return ans;
    }
    string decodeMessage(string key, string message) {
        //step1: create mapping
        unordered_map<char,char> mapping;
        char spaceCharacter = ' ';
        mapping[spaceCharacter] = spaceCharacter;

        createMapping(mapping, key);

        //step2: use mapping and decode msg
        string ans = useMapping(mapping, message);
        return ans;
    }
};
```

Q.2391 Minimum Amount of Time to Collect Garbage :

LeetCode Solution:

```
class Solution {
public:
    int garbageCollection(vector<string>& garbage, vector<int>& travel) {
        int pickP = 0;
        int travelP = 0;
        int lastHouseP = 0;
        int pickM = 0;
        int travelM = 0;
        int lastHouseM = 0;
        int pickG = 0;
        int travelG = 0;
        int lastHouseG = 0;

        for(int i=0; i<garbage.size(); i++){
            string currHouse = garbage[i];
            for(int j=0; j<currHouse.length(); j++){
                char garbageType = currHouse[j];
                if(garbageType=='P'){
                    pickP++;
                    lastHouseP=i;

                }
                if(garbageType=='M'){
                    pickM++;
                    lastHouseM=i;
                }
                if(garbageType=='G'){
                    pickG++;
                    lastHouseG=i;
                }
            }


        }
        for(int i=0; i<lastHouseP; i++){
                travelP+=travel[i];
            }
            for(int i=0; i<lastHouseM; i++){
                travelM+=travel[i];
            }
            for(int i=0; i<lastHouseG; i++){
                travelG+=travel[i];
            }

            int totalPickTime = pickP+pickM+pickG;
            int totalTravelTime = travelP+travelM+travelG;
            int totalTime = totalPickTime+totalTravelTime;

        return totalTime;
    }
};
```

Q.791. Custom Sort String :

LeetCode Solution:

```
static string str;
class Solution {
public:
    static bool cmp(char a, char b){
        return (str.find(a)<str.find(b));
    }

    string customSortString(string order, string s) {
        str=order;
        sort(s.begin(), s.end(), cmp);
        return s;
    }
};
```

Q.2125 Number of Laser Beams in a Bank :

LeetCode Solution:

```
// baki
```

Q.890. Find and Replace Pattern:

LeetCode Solution:

```
class Solution {
public:
    void normalise(string &str){
        char start ='a';
        unordered_map<char, char> mapping;

        for(int i=0; i<str.length(); i++){
            char stringChar = str[i];
            if(mapping.find(stringChar) == mapping.end()){
                mapping[stringChar]=start;
                start++;
            }
        }
        for(int i=0; i<str.length(); i++){
            char mappedChar = mapping[str[i]];
            str[i]=mappedChar;
        }
    }
    vector<string> findAndReplacePattern(vector<string>& words, string pattern) {
        vector<string> ans;
        normalise(pattern);
        for(int i=0; i<words.size(); i++){
            string currWord =words[i];
            normalise(currWord);
            if(currWord.compare(pattern)==0){
                ans.push_back(words[i]);
            }
        }
        return ans;
    }
};
```