PAT 甲级刷题实录——1016

作者：aopstudio
原文链接：https://ld246.com/article/1580354412471
来源网站：链滴

原题链接

https://pintia.cn/problem-sets/994805342720868352/problems/994805493648703488

思路

这题的关键在于使用什么数据结构去存输入的数据，以及如何计算每次通话的费用。一开始我考虑了个非常复杂的方法，发现写了一百多行代码才开了个头，觉得不太对劲，于是就上网搜索了其他人是么做的，看了之后豁然开朗。在读入各个时段的话费价格时，进行累加操作最后得出全天的话费方便续计算。另外需要定义一个存储每条通话记录的结构体。定义一个 vector 存储所有输入的记录。读以后需要对所有记录进行排序，题目的要求是对各个用户按字典顺序排序，对同一个的用户的记录按间先后排序，所以 sort 方法的第三个参数就是专门为此设计的 compare 方法。最后输出结果的时候要对所有记录亮亮进行配对看是否能配上。话费的计算则采用了一个非常巧妙的方法，即都转换为从月的 0 号 0 点 0 分（理论上的）开始总计的话费，最后将挂断时的总计话费和接通时的总计话费相即得出该次通话的话费。

代码

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

struct record    //通话记录信息
{
    string name;    //通话客户名
    int month, day, hour, minute, time, status;
    //time代表从当月0号0点0分开始（理论上的）开始经过的总时间，用于比较时间先后；status代是接通还是挂断
};
bool compare(record a, record b)
{
    return a.name == b.name ? a.time < b.time : a.name < b.name;    //将记录先按客户名字典顺，后按时间顺序排序
}
int dayToll = 0;    //一整天的费用
vector<int> tolls;    //存储每个小时的话费信息
double tollFromZero(record r);
int main()
{
    vector<string> index;    //姓名和对应的数组编号
    for (int i = 0; i < 24; i++)
    {
        int toll;
        cin >> toll;
        tolls.push_back(toll);
        dayToll += toll * 60;
    }
    int N;
    cin >> N;
```

```
    vector<record> records(N);    //存储所有通话记录
    for (int i = 0; i < N; i++)
    {
        cin >> records[i].name;
        scanf_s("%d:%d:%d:%d", &records[i].month, &records[i].day, &records[i].hour, &records[i
.minute);
        string sta;
        cin >> sta;
        if (sta == "on-line")
            records[i].status = 1;
        else
            records[i].status = 0;
        records[i].time = records[i].day * 24 * 60 + records[i].hour * 60 + records[i].minute;
    }
    sort(records.begin(), records.end(), compare);
    map<string, vector<record> > result;    //存储每个顾客对应的通话记录
    for (int i = 1; i < records.size(); i++)
    {
        if (records[i].name == records[i - 1].name&&records[i].status == 0 && records[i - 1].stat
s == 1)    //能够配对
        {
            string name = records[i].name;
            result[name].push_back(records[i - 1]);
            result[name].push_back(records[i]);
        }
    }
    for (auto it : result)
    {
        vector<record> temp = it.second;
        cout << it.first << ' ';
        printf("%02d\n", temp[0].month);
        double total = 0.0;    //总费用
        for (int i = 0; i < temp.size(); i += 2)
        {
            double onceToll;    //单次通话费用
            onceToll = tollFromZero(temp[i + 1]) - tollFromZero(temp[i]);
            printf("%02d:%02d:%02d ", temp[i].day, temp[i].hour, temp[i].minute);
            printf("%02d:%02d:%02d ", temp[i + 1].day, temp[i + 1].hour, temp[i + 1].minute);
            printf("%d ", temp[i + 1].time - temp[i].time);
            printf("$%.2lf\n", onceToll);
            total += onceToll;
        }
        printf("Total amount: $%.2lf\n", total);
    }
    return 0;
}
double tollFromZero(record r)
{
    double sumToll = dayToll * r.day + tolls[r.hour] * r.minute;    //把当月前几天的费用和当前小
分钟数的费用算了
    for (int i = 0; i < r.hour; i++)    //再算当天前几个小时的费用
    {
        sumToll += tolls[i] * 60;
    }
```

```
    return sumToll / 100;    //单位从分转换为元
}
```

原文链接：PAT 甲级刷题实录——1016