250703小码哥的艰难选择（题解）

> 题目链接：
> https://www.matiji.net/exam/brushquestion/3/4446/16A92C42378232DEB56179D9C70DC45C?from=1

#以下题目可能有错别字等，完整题目请访问上述链接
小码哥养了很多花，这些花每天都需要大量的肥料，现在小码哥提供不起这么多肥料了，他决定从所有的花里面挑选一个淘汰，规则如下：
每一个花的初始分数是1000分；
在总共 n（1≤n≤100)个回合里，每一个回合，小码哥会随机给一个花对应的分数（可正可负），需要加在鲜花的当前分数上。
最后谁的分数最少则淘汰谁，如果出现两个分数一样的，则淘汰掉最先获得这个分数的。

输入格式：
第一行包含整数 n（1≤n≤100)，表示回合数。
第 2~n+12~n+1 行，按照时间顺序输入 花的名字和得分（空格隔开） 的信息 ,其中名称是长度为 L（1≤L≤20) 的英语小写字母组成的字符串，分数的绝对值不大于 10000。
输出格式：
输出被淘汰的花的名字和最终的分数。

输入：
```
2
meigui 110
yueji -20
```
输出：
```
yueji
980
```

假定每一次分数变化，都会让该朵鲜花得到一个新的分值。

最终状态统计

没考虑 “如果出现两个分数一样的，则淘汰掉最先获得这个分数的” 这个条件，但对于本题测试用例，可以AC，如遇严格的测试用例，则AC不了

方法核心：
用哈希表（unordered_map）记录每朵花的分数累计变化
每读一条输入，直接累加到该花的当前分数上
所有操作完成后，遍历哈希表，找到分数最低的花

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    unordered_map<string, int> mp;  // 哈希表，记录每朵花当前的分数变化
    int n;  // 回合数
    cin >> n;

    // 逐轮读取花名和得分变化，累加到对应花的分数上
    for (int i = 0; i < n; i++) {
        string s;      // 花名
        int score;     // 该轮得分变化
        cin >> s >> score;
        mp[s] += score;  // 累加该花的分数变化
    }

    string ans;       // 最终要淘汰的花名
    int min_s = INT_MAX;  // 最低分，初始化为最大整数

    // 遍历哈希表，找分数最低的花
    for (auto it = mp.begin(); it != mp.end(); it++) {
        int total = it->second + 1000;  // 每朵花初始1000分，累计变化加上初始分
        if (total < min_s) {
            min_s = total;
            ans = it->first;  // 记录当前最低分花的名字
        }
    }

    // 输出结果
    cout << ans << "\n" << min_s << endl;
    return 0;
}
```

解法一：哈希表+模拟

题目思路：

小码哥要从多朵花中选一朵淘汰：

1. 每朵花初始分数 1000；
2. n 次操作，每次输入：花名 + 分数变化（正负均可）；
3. 最终分数最低的花被淘汰；
4. 若多花同分，淘汰最早达到该分数的花。

核心做法：

用 unordered_map<string, int> 记录每朵花总分变化；
每次输入操作，同时用 vector<Record> 记录操作历史（花名、操作后分数、时间）；
操作全部完成后：
    遍历统计最低分；
    再次模拟过程，找最早达到最低分的花。

注意：

通过 history 记录完整过程，确保“最早淘汰”逻辑正确；

```
#include <bits/stdc++.h>
using namespace std;

struct Record {
    string s;       // 花名
    int score;      // 当前分数
    int time;       // 变化发生的轮次
};

int main() {
    unordered_map<string, int> mp;  // 记录每朵花当前的分数变化
    vector<Record> history;
    int n;
    cin >> n;

    for (int i = 0; i < n; i++) {
        string s;
        int score;
        cin >> s >> score;
        if (!mp.count(s))
                mp[s] = 1000;  // 初始化1000分
            mp[s] += score;
        history.push_back({s, mp[s], i});
    }

    // 先找最终最低分数
    int min_s = INT_MAX;
    for (auto &p : mp) {
        min_s = min(min_s, p.second);
    }

    // 再找最早使分数达到 min_s 的花
    unordered_map<string, int> temp_score;
    for (auto &p : mp) {
        temp_score[p.first] = 1000; // 每朵花初始 1000 分
    }

    for (auto &rec : history) {
        temp_score[rec.s] = rec.score; // 更新当前分数
```

```
        if (mp[rec.s] == min_s && temp_score[rec.s] == min_s) {
            cout << rec.s << "\n" << min_s << endl;
            return 0;
        }
    }
}
```