C++方向编程题答案

答案说明：

> 大家如果对本次题目或者答案有问题，可以联系下方的出题老师答疑。
>
> **出题老师：**
>
> 选择题：吴都 qq：1226631755
>
> 代码题：时亮益 qq：569334855

第六周

day35

1、题目ID：772 年会抽奖

链接：https://www.nowcoder.com/questionTerminal/610e6c0387a0401fb96675f58cda8559

【题目解析】：

该题为经典的问题：**错排问题**

用A、B、C……表示写着 n 位友人名字的信封，a、b、c……表示 n 份相应的写好的信纸。把错装的总数为记作D(n)。假设把 **a 错装进 B 里了**，包含着这个错误的一切错装法分两类：

- b装入 A 里，这时每种错装的其余部分都与A、B、a、b无关，应有D(n - 2)种错装法。
- b 装入A、B之外的一个信封，这时的装信工作实际是把（除a之外的）n - 1份信纸 b 、 c ……装入（除B以外的）n - 1个信封A、C……，显然这时装错的方法有D(n - 1)种。

总之在 a 装入B的错误之下，共有错装法D(n - 2) + D(n - 1)种。

a装入C，装入D……的n - 2种错误之下，同样都有D(n - 1) + D(n - 2)种错装法，因此D(n) = (n - 1)[D(n - 1) + D(n - 2)]

D(n) = (n-1) [D(n-2) + D(n-1)]

特殊地，D(1) = 0, D(2) = 1.

【解题思路】：

错排的递推公式是：D(n) = (n - 1) [D(n - 2) + D(n - 1)]，也就是n - 1倍的前两项和。公式推导可以参考百度百科。通过这个递推公式可以得到在总数为n的时候，错排的可能性一共有多少种。那么要求错排的概率，我们还需要另一个数值，就是当总数为n的时候，所有的排列组合一共有多少种，那么根据排列组合，肯定是使用

$$A_n^n$$

的公式来求，也就是n的阶乘。所以结果很简单，就是用公式求出第n项的错排种类，和n的阶乘，然后两者一除，就是概率了。

【示例代码】：

```
#include <iostream>
#include <cstdio>
```

```
int main()
{
    long long d[21] = { 0, 0, 1 };  // 错排数量，预留第一项为0，配合下文中输入的n
    long long f[21] = { 1, 1, 2 };  // 阶乘
    for (int i = 3; i <= 20; i++)
    {
        d[i] = (i - 1) * (d[i - 1] + d[i - 2]); //错排的递推公式
        f[i] = i * f[i - 1]; //阶乘的递推公式
    }

    int n;
    while (std::cin >> n)
    {
        printf("%.2f%%\n", 100.0 * d[n] / f[n]); //用100.0来把结果处理成double，保留两位小数
    }
    return 0;
}
```

2、题目ID：840 抄送列表

链接：https://www.nowcoder.com/questionTerminal/286af664b17243deb745f69138f8a800

【题目解析】：

本题是在第一行的人名中，查找第二行的人名是否存在。牵涉一个全字匹配的问题。

【解题思路】：

1. 通过getiine(cin, names)方法获取第一行中的所有名字
2. 解析出第一行中的所有名字保存在unordered_set中
3. 获取第二行中的名字，检测该名字是否存在，并按照题目的要求进行输出

【示例代码】：

```
#include <iostream>
using namespace std;
#include <unordered_set>
#include <string>

int main()
{
    // 循环处理每一组测试用例
    string name;
    while (getline(cin, name))
    {
        // 将第一行中的所有名字进行拆解，保存在unordered_set中，方便后序查找
        unordered_set<string> s;
        size_t pos = 0;

        while (pos < name.size())
        {

            // 该名字使用""包含了，将该名字截取出来
```

```
            if (name[pos] == '\"')
            {
                size_t end = name.find("\"", pos + 1);
                s.insert(name.substr(pos + 1, end - pos - 1));
                pos = end + 2;//跳掉后面的双引号和逗号
            }
            else
            {
                // 该名字没有使用""包含，找到改名字的末尾后直接截取
                size_t end = name.find(",", pos + 1);
                if (end == -1)
                {
                    // 已经是最后一个名字了
                    s.insert(name.substr(pos, name.size() - pos));
                    break;
                }

                s.insert(name.substr(pos, end - pos));
                pos = end + 1; //跳掉后面的逗号
            }
        }

        // 接收第二行的名字，然后检测其是否在unordered_set中存在
        getline(cin, name);
        if (s.find(name) == s.end())
        {
            printf("Important!\n"); //没找到
        }
        else
        {
            printf("Ignore\n"); //找到了
        }
    }
    return 0;
}
```