F. 等比数

#include<bits/stdc++.h>

using namespace std;

/\*

1. 三个三位数满足 1:2:3

2. 枚举第1个三位数

如果第1个三位数是：i

第2个三位数一定是：2 \* i

第3个三位数一定是：3 \* i

检验：i、2i、3i，用到1~9的每个数

\*/

int c[20];//统计数组

int x;

bool f;

int main() {

    //枚举第1个3位数

    for(int i = 100;i <= 333;i++){

        //将3个3位数，拼成一个9位数

        x = i \* 1000000 + 2 \* i \* 1000 + 3 \* i;

        memset(c,0,sizeof(c));

        while(x != 0){

            c[x % 10]++;

            x /= 10;

        }

        //如果c数组1~9，对应的值都是1，说明满足条件

        f = true;//假设满足条件

        for(int j = 1;j <= 9;j++){

            if(c[j] != 1){

                f = false;

                break;

            }

        }

        if(f){

            cout<<i<<" "<<2 \* i<<" "<<3 \* i<<endl;

        }

    }

    return 0;

}

M. 投篮(b.cpp)

#include<bits/stdc++.h>

using namespace std;

/\*

1. 投中：V +1

2. 不中，但不扣分：T +0

3. 三不沾，扣1分：X -1

只要连续投中3个及3个以上，除了+1分以外

额外再+1分

连续性统计：统计连续的V出现的次数

\*/

int n;

string s;

int c = 0;//统计连续的V出现的次数

int ans = 0;

int main() {

    cin >> n >> s;

    for(int i = 0; i < n; i++) {

        //如果是V

        if(s[i] == 'V') {

            c++;

            ans++;//加一分

            if(c >= 3) ans++;//额外加一分

            //判断连续的V结束

            if(i == n - 1 || s[i + 1] != 'V') {

                c = 0;

            }

        } else if(s[i] == 'X') ans--;

    }

    cout << ans;

    return 0;

}

Q. 洗牌

#include<bits/stdc++.h>

using namespace std;

int a[1010],b[1010];

int n,k,x;

int main() {

    cin>>n>>k>>x;

    //将n张牌的原始顺序存入a数组

    for(int i = 1;i <= n;i++){

        a[i] = i;

    }

    //k次洗牌

    while(k--){

        //每次洗牌，将洗牌的结果存入b数组

        for(int i = 1;i <= n / 2;i++){

            //a数组的前一半，存入b数组的下标为奇数的位置

            b[2 \* i - 1] = a[i];

            //a数组的后一半，存入b数组的下标为偶数的位置

            b[2 \* i] = a[n / 2 + i];

        }

        //洗牌结束，将b数组拷贝回a数组，进行下一次洗牌

        for(int i = 1;i <= n;i++){

            a[i] = b[i];

        }

    }

    cout<<a[x];

    return 0;

}

U. 趣味填空

#include<bits/stdc++.h>

using namespace std;

/\*

1234=127

求：加号加在哪里。

思路：

1. 将=两边的数字分解出来。

x=1234

y=127

2. 尝试将x所有拆位的可能都拆一遍

123+4

12+34

1+234

\*/

string s,s1,s2;//s1代表=左侧的数字，s2代表=右侧的数字

int p;//表示=的位置

int x,y,z;

int main() {

    cin>>s;

    p = s.find("=");//1234=127

    s1 = s.substr(0,p);

    s2 = s.substr(p+1);

    z = stoi(s2);

    //cout<<s1<<endl<<s2;

    /\*

    将s1的所有拆分可能都尝试一遍

    1234

    1 234

    12 34

    123 4

    归纳法

    \*/

    string t1,t2;

    bool f = false;

    //最多拆s1.size()-1

    for(int i = 1;i <= s1.size() - 1;i++){

        t1 = s1.substr(0,i);

        t2 = s1.substr(i);

        //cout<<t1<<" "<<t2<<endl;

        x = stoi(t1);

        y = stoi(t2);

        if(x + y == z){

            cout<<x<<"+"<<y<<"="<<z<<endl;

            f = true;

        }

    }

    if(f == false){

        cout<<"Impossible!";

    }

    return 0;

}

W. 回文字符串

#include<bits/stdc++.h>

using namespace std;

/\*

如果回文字符串长度是偶数：

所有字母出现的次数都是偶数

如果回文字符串长度是奇数：

一定有一种字母出现的次数是奇数，其余是偶数

如果出现次数为奇数的字母，有0种或者1种，一定能够构成回文

\*/

int c[200];

string s;

int n;

int main() {

    cin>>n;

    while(n--){

        cin>>s;

        memset(c,0,sizeof(c));

        //计算s中每种字母出现的次数

        for(int i = 0;i < s.size();i++){

            c[s[i]]++;

        }

        //统计出现次数为奇数的字母有几种

        int x = 0;

        for(char i = 'a';i <= 'z';i++){

            if(c[i] % 2 == 1){

                x++;

            }

        }

        if(x <= 1) cout<<"Yes"<<endl;

        else cout<<"No"<<endl;

    }

    return 0;

}

**总结：**

1. 多组测试数据的问题，要注意每次先清空一些变量。

2. 多组测试数据的问题，一定要注意输出格式。

3. 变量先定义再使用。

C. 分数计数

#include<bits/stdc++.h>

using namespace std;

int n,x[110];

int cnt[110];//统计数组：队伍的编号是1~n

int c = 0;//数字连续出现次数的计数器

int main() {

    cin>>n>>x[1];

    //递推出每次的胜队

    for(int i = 2;i <= n;i++){

        x[i] = ((x[i-1] \* 3703 + 1047) % n) + 1;

    }

    for(int i = 1;i <= n;i++){

        c++;

        //连胜次数<=3次，得分 = c

        if(c <= 3) cnt[x[i]] += c;

        else cnt[x[i]] += 3;

        //判断连续相等的条件结束

        if(i == n || x[i] != x[i + 1]){

            c = 0;

        }

    }

    //求得分最多的队伍，得分是多少

    int ma = cnt[1];

    //n个队伍的编号是1~n

    for(int i = 2;i <= n;i++){

        ma = max(ma,cnt[i]);

    }

    cout<<ma;

    return 0;

}

E. 回文日期（date）

#include<bits/stdc++.h>

using namespace std;

/\*

反向构造：构造出所有的回文日期

    月日

92800829

1. 枚举所有月日的组合

2. 每次确定月日之后，反向构造出年

得到一个完整的8位数，如果该8位数在指定日期的范围内，说明回文日期的数量+1

\*/

int m[13] = {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

int s, e, x, y;

int ans = 0;

int main() {

    cin >> s >> e;

    //构造所有的回文日期

    for(int i = 1; i <= 12; i++) {

        for(int j = 1; j <= m[i]; j++) {

            x = i \* 100 + j;//后2位构造的4位数

            //利用后2位构造的4位数，算出前4位

            //i=02 j=19 => 9120

            y = j % 10 \* 1000 + j / 10 \* 100 + i % 10 \* 10 + i / 10;

            if(y \* 10000 + x >= s && y \* 10000 <= e) {

                ans++;

            }

        }

    }