

14211101_刘柘林 第六次上机解题报告

//（本次上机难度是历次最大的，大家不要丧气，连蔡大神也只做出 5 道不是，解题报告写得十分认真，重点思路及关键难点都用加粗及红色部分标注，希望能启发下还不会的同学一点）

A: jhljx 学数组

Problem Description

终于到数组啦。。呵呵呵。。

jhljx 决定开始学习数组了。。

一天老师给了他一个字符串。。字符串吖。。能吃吗？

老师让你统计字符串中每个字符的个数。是不是很简单吖。

Input

输入多组数据。

每组数据为一个字符串，保证字符串的长度小于 1000。字符串中保证只有英文字母。

Output

输出字符串中出现的字符的个数。请按照先输出小写字母的个数，再输出大写字母的个数的顺序输出。每组数据间用一个空行隔开。详细请见样例。

Sample Input

```
AbccbA
abcdeBCCDE
```

Sample Output

```
# # #
# # #
[b] [c] [A]

#
# # # # # # # # #
[a] [b] [c] [d] [e] [B] [C] [D] [E]
```

解题分析：

这道题是一道偏难的题，第一眼看到这道题感觉就是不好做，难点有二：

1. 关于字符个数的统计
2. 字符按顺序输出，并且 **竖行** 输出。

解决方案：

1. 用两个大小为 26 的数组储存大写字母和小写字母的字符个数，这样既可解决统计问题，又可以在输出的时候按字母表顺序输出。
2. 关于竖行输出的问题一直没想好=。= 总是没想出如何竖着输出，后来刘京欣的提示点醒了我，既然无法竖着输出，那就按最自然的横着一行一行输出，这样只要设置一个 max（最大的数组元素），当数组元素 < max 输出 3 个空格，else 输出 # 。

代码如下：

```
#include <iostream>           //常用头文件在上机前就写好，掌握先机！
#include <cstdio>               //
#include <string>               //
#include <cstring>              //
#include <cmath>                //
using namespace std;
```

```

int main()
{
    int a[26] = { 0 };    // 储存大写字母的统计数组
    int b[26] = { 0 };    // 储存小写字母的统计数组
    string s;             // 声明一个字符串变量 s

    while (cin >> s)
    {
        int len = s.size(); // s 的长度，写成 s.length()也可以

        for (int i = 0; i < len; i++)
        {
            if (s[i] < 97) { b[s[i] - 'A']++; } //97 为 a 的 ASCII 码
            // 如果某一位小于 a，那么为大写字母，s[i]-‘A’为该大写字母在大写字母中出现的
            // 位数，如 A 的个数储存在 b[0]中，B 的个数储存在 b[1]中，以此类推。
            else { a[s[i] - 'a']++; } //同上
        }

        int max = -1;

        for (int i = 0; i <= 25; i++)
        {
            if (max < a[i]) { max = a[i]; }

            if (max < b[i]) { max = b[i]; }
        } // 得到 52 种字符中出现次数最多的大小 max

        while (max != 0) /* 这里前往不要写成 max--，否则接下来调
        用 max 的值判断时，参加判断的 max 已经减了 1，会出奇特的 BUG ，而且及
        其难排除。=。=。。。我排除了 10 分钟才看出来*/
        {
            for (int i = 0; i <= 25; i++)
            {
                if (a[i] > 0 && a[i] < max)
                {
                    cout << " ";
                }
                else if (a[i] > 0 && a[i] >= max)
                {
                    cout << " # ";
                }
            }
            for (int i = 0; i <= 25; i++)
            {

```

```

        if (b[i] > 0 && b[i] < max)
        {
            cout << "    ";
        }
        else if (b[i] > 0 && b[i] >= max)
        {
            cout << " # ";
        }
    }
}

```

//两个 for 循环分开写，先输出小写的，在输出大写的

```

        cout << endl;
        max--;    // 这里写 max—就没问题啦
    }

    for (int i = 0; i <= 25; i++)
    {
        if (a[i] > 0)
        {
            cout << "[" << (char)(i + 'a') << "];";
        }
    }

    for (int i = 0; i <= 25; i++)
    {
        if (b[i] > 0)
        {
            cout << "[" << (char)(i + 'A') << "];";
        }
    }

    cout << endl << endl;
    memset(a, 0, sizeof(a));
    memset(b, 0, sizeof(b));
    max = 0;
// 清空数组是个好习惯
    }
}

```

PS:

1. 用数组储存不容易发生改变的值，当这些值具有某种顺序时格外有效。
2. 用--, ++ 这些符号时需警惕。
3. 不要打错字，本人把 main 写成了 mian 前 30 分钟废了 555.

B: 巫女的怀胎

Problem Description

若非技术，那便是才能。

比如 lxx 在算术方面具有卓越的才能。

已知 x 加上 x 的各个数字之和为 y ，则说明 x 和 y 是一对 cp。给出 $n(1 \leq n \leq 100000)$ 。

求 n 的最小 cp。无解输出 0。

Input

输入多组数据。（小于 100 组）

每组数据一行，为一个数 n 。($1 \leq n \leq 100000$)

Output

每组数据输出一行。为 n 的 cp。

Sample Input

216

121

Sample Ouput

198

0

解题分析：

这不是道难题，关注下题中给的范围 $n(1 \leq n \leq 100000)$ 。那么 cp 数的各位数之和最多是 $9*5=45$ ，那么 cp 数最小是 $n-45$ ，写个 for 循环就 OK 了。

代码如下：

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

int main()
{
    int n;

    while(cin >> n)
    {
        int ans = 0;
        int i;
        for(i = n - 46; i < n; i++)
        {
            int j = i;      // 用另外一个 j 贮存 i 的值，对 j 进行之后的操作

            while(j > 0)
            {
                ans += j % 10;
                j = j / 10;
            }

            if(ans + i == n)
            {
                cout << i << endl;
                break;
            }
            else
```

```

        {
            ans = 0;
        }

    }
    if (i == n)cout << "0" << endl;
}
}

```

C: 善恶的彼岸

Problem Description

You guys do not notice that we are gifted just for being humans. We are absolute predators, we do not even have any enemies. Maybe there are animals watching us and thinking that "Someday we will BEAT THEM DOWN!". 但是"Someday we will BEAT THEM DOWN!"这种格式会导致让人分不出是左引号还是右引号，所以我们需要把它变为“”。一句话：把"变为“”。

Input

输入一篇文章。

Output

输出引号转换后的文章。

Sample Input

"To be or not to be," quoth the Bard, "that is the question".

Sample Output

“To be or not to be,” quoth the Bard, “that is the question”.

Hint

注意是将英文引号转化为中文引号。

解题分析：

这不是道难题，如果有字符串数组 `char[]` 或者 `string` 的知识简直是签到题，只需要注意下左引号和右引号的区别就好，用一个标记变量就好。

代码如下：

```
#include <iostream>
#include <cstdio>
#include <string>
#include <cstring>
using namespace std;

int main()
{
    string s;
    while (cin >> s)
    {
        int len = s.size();
        bool q = true;           // 声明一个标记变量，其类型为 bool
        for (int i = 0; i < len; i++)
        {
            if (s[i] == "\"" && q)    // 如果 q 为真那么是左引号
            {
                cout << "¡°";        // 一堆乱码不要在意
                q = !q;              // 真变假
            }
            else if (s[i] == "\"" && !q) // 若为假，为右引号
            {
                cout << "¡±";
                q = !q;              // 假变真
            }
            else cout << s[i];
        }
    }
}
```



```
}
```

PS:

弱弱问下为什么我的程序不能通过编译却通过了评测??? 是编译器问题么? 自己的CB是最新的, 不能输出中文? 用 `wcout` 也没用。。。

D: jhljx 分解质因数

Problem Description

jhljx 最近在学小学数学, 老师教他分解质因数。也就是说给你一个数 n , 让你把它分解成若干个质数的乘积的形式。

Input

输入多组数据。

每组数据一行, 为一个数 n 。($1 \leq n \leq 10^6$)

Output

每组数据输出一行。为一个表达式。表达式中的数字必须按从小到大的顺序排列。(详见样例)

Sample Input

```
11
9412
```

Sample Ouput

```
11
2*2*13*181
```

Hint

注意可能木有乘号哦。。
请用 `scanf` 和 `printf` 输入输出。

解题分析：

这是一道中难题，考察对循环的深入理解。题意清晰给好评。

我已开始是想写另外一个函数的，这样就可惜实现递归判断，但是感觉太繁琐，想起 `break` 这个强大的工具，有了第二种思路。

代码 1：// 十分丑陋，纯想过题，不推荐

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

int a[168] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89,
97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193,
197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311,
313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433,
439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569,
571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683,
691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827,
829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971,
977, 983, 991, 997};
// 开了一个数组，里面为 1000 以内的质数

int f(int);    // 写一个可以递归的函数，该代码主体

int main()
{
    int n;

    while(cin >> n)
    {
        if (n == 1)cout << "1" << endl;
        else f(n);
    }
}

int f(int n)
```

```

{
    for(int i = 0; i<168; i++)
    {
        if(n == a[i])
        {
            cout << a[i] << endl;
            return 0;
        }
    }
    for(int i = 2; i * i <= n; i++)
    {
        if (n % i == 0)
        {
            cout << i << "*";
            n = n / i;
            f(n);
            break;
        }
    }
}
}

```

// 同样这里的 **break** 也十分重要, 否则会输出多个乘式。

Ps:

这个代码经测试是可以的, 但就是 **WA** 了, 不明所以, 难道刘京欣说 $n < 10^6$ 是骗我的??

代码 2:

```

#include <cstdio>

int main()
{
    int n, i;

    while(scanf("%d", &n) != EOF)
    {

        for(i = 2; i <= n; i++)
            while(n != i)
            {
                if(n % i == 0)
                {
                    printf("%d*", i);

```

```

        n = n / i;
    }
    else
    {
        break;    // 调出循环，回到 for 循环，这个技巧性十分强
    }
}

printf("%d\n", n);
}
}

```

E: jhljx 学排列组合

对于这道题不想再多说什么了，看了都想呕。

当然 借鉴意义还是有的：

- 1.不要忽略细节，细节很重要，比如那个 YeS。
- 2.不卡内存的情况下，尽量把数据开得大一点，减少不必要的错误

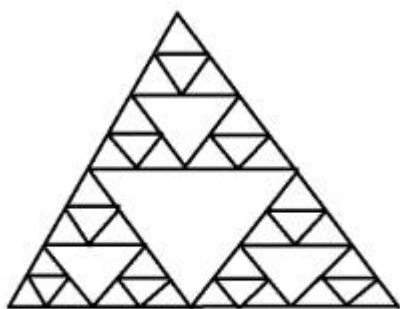
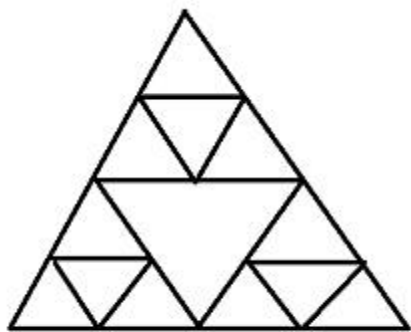
F: jhljx 学画画

Problem Description

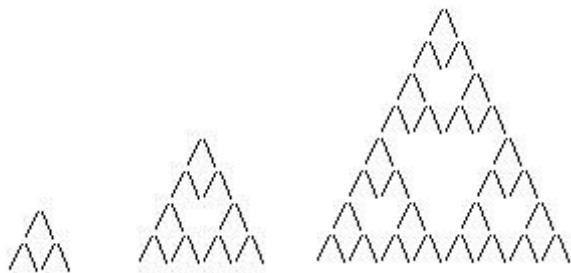
jhljx 是一个爱好广泛的人。最近他迷恋上了画画。还记得上次他教松辰学妹学画画。这次他决定自己动手画画啦。。

真是好棒耶。。~\(\geq \nabla \leq)/~啦啦啦

他要画的图形是这样的。。



但是为了简化，我们将图形变成了这样。



Input

输入多组数据。

每组数据输入一个 n ，表示图形的行数。(保证 n 为 2,4,8,16.....，且 $n \leq 10000$)

Output

输出正确的图形。

Sample Input

8

Sample Output

解题分析:

这道题是一道难题，题意十分清晰，通过自己画图 and 观察样例，可以很快得到其递推的形式，但难点也就出来了，就算你知道是怎么一回事，也有一种无从下手的感觉，上半层的空格实在不好处理，介于文本的输出是按行输出的，一层一层打也不现实。

于是!!!

技巧不行的话，我们回头看看题中的一个条件： **$n \leq 10000$**

也就是说，最多我们只要输出 n 为 2^{13} 的情况，

而对于这种有限的情况，而且情况比较少，那么不妨使用枚举法!!!

思路就出来了：我们既然不好直接按行输出，那么不妨转化下思路，把它想成一个 $n \times 2n$ 的表格，只要输出每一个格子的样子就行了，而通过我们找到的规律，通过递归是可以实现的。

代码如下：（借用胡晟大神的代码，不要在意）

```
#include <stdio>
```

```
using namespace std;
```

char map[10010][20010]; //开一个最大的数组，之所以用 char 是由于 char 是一字节的，占用内存最少，用 short 会爆内存(你说 int? =。=想多了)

```

void dodo(int, int, int);          // 递归函数

int main()
{
    int n, i, j;
    while (scanf("%d",&n)!=EOF)
    {
        for (i = 1; i <= n; i++)
            for (j = 1; j <= n * 2; j++)
                map[i][j] = '0';          // 将数组清空
        dodo(n, 1, 1);
        for (i = 1; i <= n; i++)
        {
            for (j = 1; j <= 2 * n; j++)
                if (map[i][j] == '1') printf("/");          // 若 为 1 输出/
            else
                if (map[i][j] == '2') printf("\\");          // 若 为 2 输出\
            else printf(" ");
            printf("\n");
        }
    }
}

void dodo(int n, int x, int y)
{
    if (n == 1)                          // 最开始 n=1 的情况，奠基
    {
        map[x][y] = '1';
        map[x][y + 1] = '2';
    }
    else
    {
        dodo(n / 2, x, y + n / 2);          // 观察得到的普适性递推式
        dodo(n / 2, x + n / 2, y);
        dodo(n / 2, x + n / 2, y + n);
    }
}

```

PS:

对于这题，我想到了汉诺塔，都是考察的递归，而递归是相当符合人的思维模式的，十分

自然的一种写法（缺点是容易堆栈溢出，效率低），写递归体的时候不要考虑太多细节，只要把握一点是 **n 到 n-1 的关系** 就好，牢记这一点，就不会进入思维误区。

G: allkill 冰点

Problem Description

A human baby when will they find out?
That at a point they are born We are
winners of Earth.

给定从 2000 年 1 月 1 日逝去的天数，
请输出这一天的具体日期。

Input

多组数据，每组一个整数 $n(n \geq 0)$ 保证结果不超过 9999 年（保证数据量小于 $1e6$ ）

Output

输出具体日期，格式为"year-month-day dayofweek"
需要前导 0

Sample Input

1
30
365
366

Sample Ouput

2000-01-02 Sunday
2000-01-31 Monday
2000-12-31 Sunday
2001-01-01 Monday

Hint

"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"

难题，慎入

解题思路：

由于这题没有人做出来（我现在依然是 TLE），我也不好说自己思路一定好或者正确什么的，想挑战的同学就不要看下去了=。=，姑且在这里抛砖引玉。

这道题要解决两个问题，一是年月日的判断，二是星期的判断，后者不难，%7 处理即可，难的是前者。

而前者又难在两点：

1. 闰年
2. 大月，小月的判断。

我们之前做过类似的，第四次上机 F 题（掷石之人 - 2014 级 C++第四次上机 - OJ3RD <http://acm.buaa.edu.cn/contest/160/problem/F/>）就是要我们判断星期，而给的数据是年月日，相当于说，这道题是前者的**逆题**，思路上有借鉴之处。

我过 F 题是写了一个给年月日算出那天离 1 年 1 月 1 日有多少天的算法，反过来我们可以重用这个函数，先确定大致范围，逐步精细缩小范围，最终确定 answer!!!

代码如下：

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

int day[12] = { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334 } ; // 常量数组，分别为过 1
个月，2 个月。。。。的天数
int s1(int, int, int); // 计算 y.m.d 距离 1.1.1 逝去的天数
void compare(int &, int &, int &, int, int, int); // 比较两个日期过去的天数是否相等

int main()
{
```

```

string s[7] = { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
"Sunday" };          // 开一个判断星期的数组，可以看做二维的。
int n;

```

```

while(scanf("%d", &n) != EOF)
{
    int passedDay = n + 730120;          // 730120 是 2000.1.1 距离 1.1.1 过去的天数
    int startYear = passedDay / 366;     // 最小年份
    int maxYear = passedDay / 365;      // 最大年份

```

```

    int year, month , day;
    compare(year, month, day, startYear, maxYear, passedDay);

```

```

    //cout << year << " " << month << " " << day << endl;          // 测试中间变量的值

```

```

    printf("%d-", year);

```

```

    if((month == 4 || month == 6 || month == 9 || month == 11) && (day == 31))
    {

```

```

        month++;

```

```

        day = 1;
    }

```

// 这里就不解释了=。= 都是一些输出的小细节，关键在于前面那两个函数!!!

```

    if(month == 2)
    {

```

```

        if((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
        {

```

```

            if(day == 30 || day == 31)
            {

```

```

                month = 3;

```

```

                day = day - 29;
            }
        }
    }

```

```

    else
    {

```

```

        if(day == 29 || day == 30 || day == 31)
        {

```

```

            month = 3;

```

```

            day = day - 28;
        }
    }
}

```

```

        if(month < 10) {printf("0%d-", month); }
        else { printf("%d-", month);}

        if(day < 10) {printf("0%d ", day); }
        else { printf("%d ", day);}

        cout << s[(n + 5) % 7] << endl;

        year = month = day = 0;
    }
}

int s1(int y, int m, int d)
{
    int sum = 0;
    sum += (y - 1) * 365;
    sum += day[m - 1];
    sum += d;

    sum += y / 4 + y / 400;
    sum -= y / 100;

    if(((y % 4 == 0 && y % 100 != 0) || y % 400 == 0) && (m == 1 || (m == 2))) // 闰年的判断
    {
        sum--;
    }

    return sum;
}

```

void compare(int &i, int &j, int &t, int _start, int _end, int b) // 这里的 i, j, t 是按应用传递，下面对它们的修改会影响到传进来的值，也就是 year, month, day, 变量名怪怪的，一开始就没写好，之后又不想改，以后一定注意

```

{
    int q = 1; // 标记是否结束

    for (i = _start; i <= _end && q; i++)
    {
        int i_panduan = s1(i, 1, 1); // 用的变量名就是判断的意思，减少 s 函数的调用，可读性也变差了，可以直接三个 for 循环，但是效率变低很多
        while (i_panduan + 366 <= b)
        {
            if (((i % 4 == 0 && i % 100 != 0) || i % 400 == 0))
            {

```

```

        i_panduan += 366;
    }
    else i_panduan += 365;
    i++;
}
for (j = 1; j <= 12 && q; j++)
{
    while (i_panduan + 31 <= b)
    {
        i_panduan += day[j];
        j++;
        if (j == 2 && ((i % 4 == 0 && i % 100 != 0) || i % 400 == 0))
        {
            i_panduan++;
        }
    }
    for (t = 1; t <= 31 && q; t++)
    {
        if (s1(i, j, t) == b)
        {
            q = 0; // 结束三个 for 循环，但由于 i, j, t 之后还会再
            // 自加一次，所以下面要把它们自减 1
            i--;
            j--;
            t--;
        }
    }
}
}
}

```

结束语：

1.

代码的可读性与效率呈反比，但好的命名规则可以增加可读性（C++编程命名规范_Fly-Hcc_新浪博客

http://blog.sina.com.cn/s/blog_a3a8d0b1010100uw.html）

2.

做题前理清思路,可以写好伪代码,这不会拖慢你的速度,反而使你思路清晰,利于排除 BUG,欲速则不达!

3

运用自减自加符请慎重,多用常数数组,固定值的变量,一段代码运行完后请清空代码中的变量……等等。

学计算机要从娃(WA)娃(WA)抓起