A 分子量

时间限制: 1000 ms 内存限制: 65536 kb
总通过人数: 922 总提交人数: 1235

题目描述

给出一个分子式（不带括号），求相对分子质量。本题只包含 4 种原子，分别为 C，H，O，N，相对原子质量分别为 12.01,1.008,16.00,14.01,（单位：g/mol）。

输入

输入一个不带括号的分子式，字符串长度不超过 100，分子式只有大写字母。

输出

输出分子式的相对分子质量，小数点后保留三位。

输入样例

```
C2H5OH
```

输出样例

```
94.108
```

考察知识点：

字符串处理

解题思路：

将输入的串从头到尾扫描，遇到字母，则进一步扫描后面的数字的区间，进行字符串到整数的转换，再乘以其的原子质量，最后累加到 sum 中即可。
注意两个字母相邻的情况，直接累加原子质量到 sum。
代码：

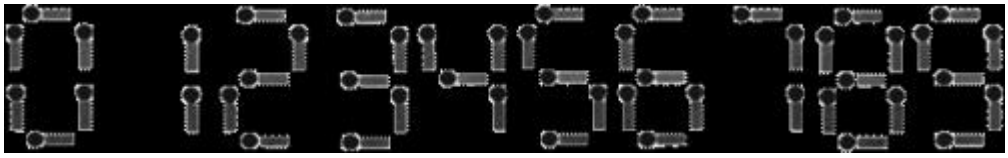
```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
char s[100];
int main () {
    scanf("%s", s);
    int i;
    int sum = 0;
    double m = 0, n = 0;
    for (i = 0; i < strlen(s); i++) {
        if (isupper(s[i])) {
            switch (s[i]) {
                case 'C': n = 12.01; break;
                case 'H': n = 1.008; break;
                case 'O': n = 16.00; break;
                case 'N': n = 14.01; break;
            }
            sum = 0;
        }
        // 字符串转整数
        while (isdigit(s[i])) {
            sum = sum * 10 + s[i] - '0';
            if (!isdigit(s[i + 1]))
                break;
            i++;
        }
        if (sum != 0)
            m += sum * n;
        else if (!isdigit(s[i + 1]))
            m += n;
    }
    printf("%.3f\n", m);
    return 0;
}
```


时间限制：1000ms　内存限制：65536kb

通过率：916/1033 (88.67%)　正确率：916/2541 (36.05%)

题目描述

给你 n 根火柴棍，你可以拼出多少个形如“A+B=C”的等式？等式中的 A、B、C 是用火柴棍拼出的整数（若该数非零，则最高位不能是 0）。用火柴棍拼数字 0-9 的拼法如图所示：



注意：

1. 加号与等号各自需要两根火柴棍
2. 如果 A≠B，则 A+B=C 与 B+A=C 视为不同的等式（A、B、C>=0）
3. n 根火柴棍必须全部用上

输入

输入为一个整数 n（0<=n<=24）

输出

输出表示能拼成的不同等式的数目。

输入样例

```
14
```

输出样例

```
2
```

说明

14 根火柴棍能拼成的两个等式是 `0+1=1` 和 `1+0=1`。

解题思路

本题是求给定数目的火柴棍能拼成多少个等式，我们发现题目规定最多用到 24 根火柴，于是我们想到可以用穷举的思路来解题，稍微观察就可以发现最多可以穷举到的式子是 1111+1=1112（实际上这个式子要用到 24 根）。所以我们从 0~1111 来枚举每一个数，用一个函数 fun 来计算某个数需要多少根火柴，就有了 if(fun(a)+fun(b)+fun(c)==m-4) sum++;
我们来看一下 fun 函数，fun 函数把数字 x 的每一位拆开然后计算该位用到的火柴数，加在一起。

标程

```
#include <stdio.h>
int fun(int x)       //用来计算一个数所需要的火柴棍总数
{
      int num=0;   //用来计数变量
      int f[10]={6,2,5,5,4,5,6,3,7,6};     //用一个数组记录 0~9 数字所需的火柴棍数
      while(x/10!=0)         // x 除以 10 不等于 0 的话，说明该数至少有两位
      {
            num+=f[x%10];     //加上该位火柴棍数
            x=x/10;
      }
      num+=f[x];         //加上最高位的火柴棍数
      return num;
}
int main()
{
       int a,b,c,m,sum=0;
       scanf("%d",&m);              //火柴棍总个数
       for(a=0;a<=1111;a++)    //开始枚举
       {
             for(b=0;b<=1111;b++)
             {
                   c=a+b;
                   if(fun(a)+fun(b)+fun(c)==m-4)    //去掉+和=
                         sum++;
```

```
            }
        }
        printf("%d",sum);
    return 0;
}
```

C 整数划分

时间限制：5000ms　内存限制：65536kb

通过率：828/927 (89.32%)　正确率：828/1442 (57.42%)

题目描述

一个整数 n 可以有多个划分方案，如 n=5 时，划分方案有以下 7 种：
5=5,
5=4+1,
5=3+2,
5=3+1+1,
5=2+2+1,
5=2+1+1+1,
5=1+1+1+1+1。
现在给你一个整数 n，请你求出 n 有多少种划分方案。
对于两种划分方案，当且仅当**两划分中的各个数（不考虑顺序）对应相等**时称这两个方案相同。

如对应 n=5,5=2+1+1+1 和 5=1+2+1+1,5=1+1+2+1 都是同一个方案。

输入

多组数据。

第一行为一个正整数 T(T<10)，为数据的组数。
对于每组数据，为一行一个正整数 n(0<n<=100)，为划分的整数。

输出

对于每组数据，输出一行一个整数 count，为划分方案数。

输入样例

```
2
5
9
```

输出样例

```
7
30
```

考察知识点

```
循环，难度 4
```

解题思路

考虑寻找递归式来求解答案，令 f(n,m)为对整数 n 进行划分，划分中所有项都不大于 m 的方案数。
首先考虑边界条件，当 n=1 或 m=1 时，不难发现 f(n,m)=1，之后我们要做的就是寻找其他递归关系使 n，m 向 1 缩小；
由于 m 是对划分的一个限制，它与 n 的关系大小也会影响答案，当 n<m 时，显然划分的任何一项都不会超过 n，所以 f(n,m)=f(n,n);
当 n==m 时，一种方案是“n=n”这个划分，其他的方案就等价于 f(n,m-1)了，因此除了这种方案其他方案的各项最大都达不到 n(也就是 m)，所以此时 f(n,m) = f(n,m-1)+1；
当 n>m 时，考虑两类划分，一类含有 m，一类不含有 m，第一类的数量为 f(n-m,m)(先拿出一个 m)，第二类的数量为 f(n,m-1)，所以此时 f(n,m) = f(n,m-1)+f(n-m,m)。
综上，我们将 f(n,m)所有情况下的递归方法都找到了，我们要找的答案就是 f(n,n)，递归求解即可。
此外，这样的递归程序在 n=100 的规模下运行速度就不理想了，原因是大量的重复调用。解决这个的方法有记忆化搜索、改写为非递归实现的递推等，大家可以自行搜索学习。

```
#include<stdio.h>
int n,T;
int f(int n,int m){
    if (n == 1)
        return 1;
    else if (m == 1)
```

```
            return 1;
        else if (n < m)
            return f(n,n);
        else if (n == m)
            return 1 + f(n,m - 1);
        else
            return f(n,m - 1) + f(n - m,m);
}
int main(){
        scanf("%d",&T);
        while(T--){
            scanf("%d",&n);
            printf("%d\n",f(n,n));
        }
        return 0;
}
```

D 傻傻 Aqi 的递归汉诺塔

时间限制：1000ms　内存限制：32MB

通过率：985/1080 (91.20%)　正确率：985/1931 (51.01%)

题目描述

傻傻 Aqi 买了一个玩具，叫作汉诺塔。就是一块平板上有三根杆，最左边的杆上自上而下、由小到大顺序串着由若干个圆盘构成的塔。

目的是将最左边杆上的盘全部移到右边的杆上，条件是一次只能移动一个盘，且不许大盘放在小盘的上面。现在 Alice 追加了玩具的玩法，在上面规则的基础上，还规定不许直接从最左（右）边移到最右（左）边。

然后 Aqi 开始思考，假如现在有 n 个圆盘，那至少要多少次移动才能把这些圆盘从最左边移到最右边？傻傻 Aqi 想到了可以编程来解决却不知从何入手，这时 Alice 说可以用递归的思想来很简单的解决这个问题。你们能帮傻傻 Aqi 来编写这个程序吗？

输入

多组数据输入；

每组输入一个正整数 n，为圆盘的个数。

输出

对于每组输入，输出最少的移动次数。

输入样例

```
1
3
10
```

输出样例

```
2
26
59048
```

解题思路

与原始的汉诺塔问题不同，这里对圆盘的移动做了更多的限制，即每次只允许将圆盘移动到中间柱子上或者从中间柱子上移出，而不允许由第一根柱子直接移动圆盘到第三根柱子。

在这种情况下，我们考虑 K 个圆盘的移动情况。为了首先将初始时最底下、最大的圆盘移动到第三根柱子上，首先需要将其上的 K-1 个圆盘移动到第三根柱子上，而这恰好等价于移动 K-1 个圆盘从第一根柱子到第三根柱子。当这一移动完成以后，第一根柱子仅剩余最大的圆盘，第二根柱子为空，第三根柱子按顺序摆放着 K-1 个圆盘。我们将最大的圆盘移动到此时没有任何圆盘的第二根柱子上，并再次将 K-1 个圆盘从第三根柱子移动到第一根柱子，此时仍然需要移动

K-1 个圆盘从第一根柱子到第三根柱子所需的移动次数(第一根柱子和第三根柱子等价)，当这一移动完成以后将最大的圆盘移动到第三根柱子上，最后将 K-1 个圆盘从第一根柱子移动到第三根柱子上。

若移动 K 个圆盘从第一根柱子到第三根柱子需要 F[K]次移动，那么综上所述 F[K]的组成方式为，先移动 K-1 个圆盘到第三根柱子需要 F[K-1]次移动，再将最大的圆盘移动到中间柱子需要 1 次移动，然后将 K-1 个圆盘移动回第一根柱子同样需要 F[K-1]次移动，移动最大的盘子到第三根柱子需要 1 次移动，最后将 K-1 个圆盘也移动到第三根圆盘需要 F[K-1]次移动，这样 F[K]=3*F[K-1]+2。即从第一根柱子移动 K 个圆盘到第三根柱子，需要三次从第一根柱子移动 K-1 个圆盘到第三根柱子，外加三次对最大圆盘的移动。若函数 F(x)返回移动 x 根柱子所需要的移动次数，那么其递归方式为 3*F(x-1)+2。

同时我们要确定递归的出口。当 x 为 1 时，即移动一个盘子从第一根柱子移动到第三根柱子，其所需的移动次数是显而易见的，为 2。即当函数的参数为 1 时直接返回 2。

这样我们就确定了解决该问题的递归函数。

代码

```
#include<stdio.h>

long long digui(int num){
    if(num==1){
        return 2;
    }else{
        return 3*digui(num-1)+2;
    }
}

int main(){
    int n;
    while(scanf("%d",&n)!=EOF){
        printf("%lld\n",digui(n));
    }
    return 0;
}
```

E 分数相加与化简

时间限制：1000ms　内存限制：65536kb

通过率：1311/1381 (94.93%)　正确率：1311/2365 (55.43%)

题目描述

给你 2 个分数，求他们的和，并要求和为最简形式

输入

输入首先包含一个正整数 T（T<=1000），表示有 T 组测试数据，然后是 T 行数据，每行包含四个正整数 a,b,c,d（0<a,b,c,d<1000），表示两个分数 a/b 和 c/d

输出

对于每组测试数据，输出两个整数 e 和 f，表示 a/b+c/d 的最简化结果是 e/f，每组输出占一行

输入样例

```
2
1 4 5 6
1 2 1 10
```

输出样例

```
13 12
3 5
```

HINT

上次上机的欧几里得算法求 GCD（最大公约数）

考察知识点

函数思想；求最小公倍数和最大公约数,难度 4

解题思路

这道题的解法和数学上进行两个分式相加与化简的步骤一样，首先需要进行分子分母通分，然后找到分子分母的最大公约数，接着将分子分母分别除以最大公约数即可得到最简分式；难度在于调用函数求最大公约数，但认真做上次上机题目的同学应该知道怎样求最大公约数，所以问题不大

参考代码

```
#include <cstdio>

int gcd(int a, int b) {
	return b ? gcd(b, a % b) : a;
}

int main()
{
	int T;
	scanf("%d", &T);
	while (T --) {
		int a, b, c, d;
		scanf("%d%d%d%d", &a, &b, &c, &d);
		int up = a * d + c * b;
		int down = b * d;
		int g = gcd(up, down);
		up /= g;
		down /= g;
		printf("%d %d\n", up, down);
	}
	return 0;
}
```

F Tarpe 酋长的代理人

时间限制：1000ms　内存限制：65536kb

通过率：752/899 (83.65%)　正确率：752/3757 (20.02%)

题目描述

Tarpe 酋长认为，生命在于工作，所以酋长很少休假，休息日一般也会保持工作。

酋长休假时，需要找到合适的代理人来处理部落日常事务，出于安全考虑，代理人将在 Tarpe 酋长的几个好朋友中产生。

部落的传统是每周的周一、周二、周五以及周末为工作日，酋长的几个好朋友都很乐意帮助酋长，以下是他们的日常安排：

Matt:某 985 涉密学校学生，每天满课，但是在每年的 1,2,7,8 月份放假 。
Fate:某研究单位工作人员， 工作日上班，非工作日休息 。
Shana：自由职业（宅）， 工作日一般在家看番（休息），但是非工作日会拼命赶 ddl 。

三个人只有在休息时才能帮助酋长工作。

Tarpe 酋长想知道在某一日期休假时代理人的候选名单。

输入

多组数据，
每组数据 1 行，
每行一个日期，格式为 yyyymmdd

输出

对于每行数字，输出所有候选人，如果存在多个候选人，按照 Matt，Fate， Shana 的顺序输出，两个名字之间用空格隔开，输出之后需要换行。

输入样例

```
19991001
20180201
```

输出样例

```
Shana
Matt Fate
```

HINT

部落使用的历法是地球公历，即格里高利历，保证当前时间线在公元 1582 年 10 月 15 日之后。
没有思路的同学记得看看 Zeller formula

考察知识点

循环控制和判断，Zeller formula,难度系数 4

解题思路

一开始因为酋长的疏忽，导致数据出现了一点问题，影响了大家的上机体验，十分抱歉。
这道题考察的就是对 Zellar formula 日期转换成星期的理解。
但是，这个公式有如下适用条件：
1. 日期必须在 1582 年 10 月 4 日之后
2. 如果月份是 1 月或 2 月需要特殊考虑，月份加 12，年份-1
第二条就是这道题目的坑点，本题源自课堂上 ppt，希望同学们对 ppt 上的例题一定要掌握，

AC 代码：

```
#include <cstdio>

int main()
{
    int year, month, day;
    while (~scanf("%4d%2d%2d", &year, &month, &day)) {
        if (month == 1 || month == 2 || month == 7 || month == 8) {
```

```
            printf("Matt ");
        }
        if (month < 3) {
            month += 12;
            -- year;
        }
        int c = year / 100;
        int y = year % 100;
        int m = month;
        int d = day;
        int w = (c / 4 - 2 * c + y + y / 4 + (13 * (m + 1) / 5) + d - 1) % 7;
        w = (w + 7) % 7;
        if (w == 3 || w == 4) {
            printf("Fate\n");
        } else {
            printf("Shana\n");
        }
    }
    return 0;
}
```

G 灯，等灯等灯

时间限制: 1000 ms 内存限制: 65536 kb

总通过人数: 1067 总提交人数: 1160

题目描述

有 n 盏灯，编号为 1～n，第 1 个人把所有灯打开，第 2 个人按下所有编号为 2 的倍数的开关（这些灯将被关掉），第 3 个人按下所有编号为 3 的倍数的开关（其中关掉的灯被打开，开着灯将被关闭），依此类推。一共有 k 个人，问最后有哪些灯开着？

输入

两个整数 n 和 k，用空格分开，k≤n≤1000

输出

输出开着的灯编号，用空格分开。

输入样例

```
7 3
```

输出样例

```
1 5 6 7
```

坑点：

不要默认灯一开始是全开的，因为人数可能是 0

解题思路：

用下标为 i 的数组代表第 i 盏灯。

对于每个人，每次都枚举全部灯，如果灯的编号是人的编号的倍数则将灯的状态反转。

最后枚举输出状态是开的灯。

代码：

```
#include <stdio.h>
#include <string.h>
#define MAXN (1000 + 10)
int a[MAXN];
int main() {
    int i, j, n, k, first = 1;
    scanf("%d%d", &n, &k);
    memset(a, 0, sizeof(a));
    for(i = 1; i <= k; i++)
        for(j = 1; j <= n; j++)
            if(j % i == 0) a[j] = !a[j];

    for(j = 1; j <= n; j++)
```

```
            if(a[j]) printf("%d ", j);
        printf("\n");
        return 0;
}
```

H 蛇形方阵

时间限制：1000ms　内存限制：65536kb

通过率：546/659 (82.85%)　正确率：546/1142 (47.81%)

题目描述

在边长为 n 的方阵里从右上角顺时针螺旋填入 1,2,...,n∗n,要求填成蛇形。例如 n=4 时方阵为：
10 11 12 1
9 16 13 2
8 15 14 3
7 6 5 4

输入

一个数 n(n<=100)

输出

输出蛇形方阵，一行中每两个数直接用空格隔开。

输入样例

```
3
```

输出样例

```
7 8 1
```

```
6 9 2
5 4 3
```

考察知识点

```
循环，难度 6
```

解题思路:

按照题目要求的方向来进行填数,起始点在(0,n-1),首先向下填,如果没有越界且当前没有被填过(元素为 0)就横坐标++,并且填数.同样的方法再向左,上右,三个方向填数,直到填完 n*n 个数.

```
#include <stdio.h>
main() {
    int n, a[101][101] = {0}, x, y, c = 0;
    scanf("%d", &n);
    x = 0; y = n - 1;
    c = a[x][y] = 1;
    while(c < n * n) {
        while(x + 1 < n && !a[x + 1][y])//向下
            a[++x][y] = ++c;
        while(y - 1 >= 0 && !a[x][y - 1])//向左
            a[x][--y] = ++c;
        while(x - 1 >= 0&& !a[x - 1][y])//向上
            a[--x][y] = ++c;
        while(y + 1 < n && !a[x][y + 1])//向右
            a[x][++y] = ++c;
    }
    for (x = 0; x < n; x++) {
        for (y = 0; y < n; y++)
            printf("%d ", a[x][y]);
        printf("\n");
    }
    return 0;
}
```