【PREV-1】核桃的数量

题目

题目描述
小张是软件项目经理，他带领3个开发组。工期紧，今天都在加班呢。为鼓舞士气，小张打算

1. 各组的核桃数量必须相同

2. 各组内必须能平分核桃（当然是不能打碎的）

3. 尽量提供满足1,2条件的最小数量（节约闹革命嘛）

输入格式
输入包含三个正整数a, b, c，表示每个组正在加班的人数，用空格分开（a,b,c<30）

输出格式
输出一个正整数，表示每袋核桃的数量。

样例输入
2 4 5

样例输出
20

解析

求最大公约数就行了

c++代码

```cpp
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstdio>
#include <vector>
#include <cmath>
#include <ctime>
using namespace std;
```

```
int gcd(int a,int b){
    int c=a,d=b,t;
    while(d){
        t=c%d;
        c=d;
        d=t;
    }
    return a*b/c;
}

int main(){
    int a[3],b;
    for (int i=0;i<3 ;i++ )
        cin >> a[i];
    sort(a,a+3);
    b=gcd(a[2],a[1]);
    cout << gcd(max(a[0],b) , min(a[0],b)) << endl;
    return 0;
}
```

【PREV-2】打印十字图

题目

题目描述
小明为某机构设计了一个十字型的徽标（并非红十字会啊），如下所示：

```
..$$$$$$$$$$$$..
..$............$..
$$$.$$$$$$$$$.$$$
$...$.......$...$
$.$$$.$$$$$.$$$.$
$.$...$...$...$.$
$.$.$$$.$.$$$.$.$
$.$.$...$...$.$.$
$.$.$.$$$$$.$.$.$
$.$.$...$...$.$.$
$.$.$$$.$.$$$.$.$
$.$...$...$...$.$
$.$$$.$$$$$.$$$.$
$...$.......$...$
$$$.$$$$$$$$$.$$$
..$............$..
..$$$$$$$$$$$$..
```

对方同时也需要在电脑dos窗口中以字符的形式输出该标志，并能任意控制层数。

输入格式
一个正整数 n（n<30）表示要求打印图形的层数。

输出格式
对应包围层数的该标志。

样例输入
1

样例输出
```
..$$$$$..
..$...$..
$$$.$.$$$
$...$...$
$.$$$$$.$
$...$...$
$$$.$.$$$
..$...$..
```

```
..$$$$$..
```

解析

水题 循环输出就行了

c++代码

```
#include <set>
#include <cstdio>
#include <cmath>
#include <cctype>
#include <cstring>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <vector>
#include <stack>
#include <queue>
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=150;
char c[maxn][maxn];

int main(){
    ios::sync_with_stdio(false);
    int n=5;
    int t;
    cin >> t;
    n=n+t*4;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            c[i][j]='.';
    for(int i=1;i<=t+1;i++){
        for(int j=i*2;j<n-i*2;j++){
            c[(i-1)*2][j]='$';
            c[n-(i-1)*2-1][j]='$';
            c[j][(i-1)*2]='$';
            c[j][n-(i-1)*2-1]='$';
        }
```

```
            c[(i-1)*2+1][i*2]='$';
            c[n-(i-1)*2-2][i*2]='$';
            c[i*2][(i-1)*2+1]='$';
            c[i*2][n-(i-1)*2-2]='$';
            c[i*2][(i-1)*2+2]='$';
            c[i*2][n-(i-1)*2-3]='$';

            c[(i-1)*2+1][n-1-i*2]='$';
            c[n-(i-1)*2-2][n-1-i*2]='$';
            c[n-1-i*2][(i-1)*2+1]='$';
            c[n-1-i*2][n-(i-1)*2-2]='$';
            c[n-1-i*2][(i-1)*2+2]='$';
            c[n-1-i*2][n-(i-1)*2-3]='$';
        }
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++)
                cout << c[i][j];
            cout << endl;
        }
        return 0;
    }
```

【PREV-3】带分数

题目

题目描述
100 可以表示为带分数的形式：100 = 3 + 69258 / 714。

还可以表示为：100 = 82 + 3546 / 197。

注意特征：带分数中，数字1~9分别出现且只出现一次（不包含0）。

类似这样的带分数，100 有 11 种表示法。

输入格式
从标准输入读入一个正整数N （N<1000*1000）

输出格式
程序输出该数字用数码1~9不重复不遗漏地组成带分数表示的全部种数。

注意：不要求输出每个表示，只统计有多少表示法!

样例输入
100

样例输出
11

解析

全排列再暴力解决就行了。

c++代码

```
#include <set>
#include <cstdio>
#include <cmath>
#include <cctype>
#include <cstring>
#include <cstdlib>
#include <iostream>
#include <iomanip>
```

```cpp
#include <sstream>
#include <vector>
#include <stack>
#include <queue>
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;
void fun(int N,int a[],int &cnt){
    long long x=0,y,z;
    for(int i=1;i<8;i++){
        x=x*10+a[i-1];
        y=0;
        for(int j=i;j<8;j++){
            y=y*10+a[j];
            z=0;
            for(int k=j+1;k<9;k++)
                z=z*10+a[k];
            if(y%z!=0)
                continue;
            if(x+y/z==N){
                //cout << N << ' ' <<x <<' '<< y<<' ' << z << e
ndl;
                cnt++;
            }
        }
    }
}
int main(){
    ios::sync_with_stdio(false);
    long long N;
    int a[10],cnt=0;
    for(int i=0;i<10;i++)
        a[i]=i+1;
    while(cin >> N){
        do{
            fun(N,a,cnt);
        }while(next_permutation(a,a+9));
        cout << cnt << endl;
    }
    return 0;
}
```

【PREV-5】错误票据

题目

题目描述
某涉密单位下发了某种票据，并要在年终全部收回。

每张票据有唯一的ID号。全年所有票据的ID号是连续的，但ID的开始数码是随机选定的。

因为工作人员疏忽，在录入ID号的时候发生了一处错误，造成了某个ID断号，另外一个ID重

你的任务是通过编程，找出断号的ID和重号的ID。

假设断号不可能发生在最大和最小号。

输入格式
要求程序首先输入一个整数N(N<100)表示后面数据行数。

接着读入N行数据。

每行数据长度不等，是用空格分开的若干个（不大于100个）正整数（不大于100000），请

每个整数代表一个ID号。

输出格式
要求程序输出1行，含两个整数m n，用空格分隔。

其中，m表示断号ID，n表示重号ID

样例输入
```
6
164 178 108 109 180 155 141 159 104 182 179 118 137 184 115 124 125
172 189 127 107 112 192 103 131 133 169 158
128 102 110 148 139 157 140 195 197
185 152 135 106 123 173 122 136 174 191 145 116 151 143 175 120 161
149 138 142 146 199 126 165 156 153 193 144 166 170 121 171 132 101
113 130 176 154 177 120 117 150 114 183 186 181 100 163 160 167 147
```

样例输出
```
105 120
```

解析

读入数据再sort排序就行了。注意输入格式 建议使用stringsteam

c++代码

```
#include <set>
#include <cstdio>
#include <cmath>
#include <cctype>
#include <cstring>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <vector>
#include <stack>
#include <queue>
#include <deque>
#include <string>
#include <algorithm>
using namespace std;
const int maxn=100000+10;

int main(){
    ios::sync_with_stdio(false);
    int n,m,a,b;
    cin >> n;
    string s;
    vector<int> x;
    getline(cin,s);
    while(n--){
        getline(cin,s);
        stringstream ss(s);
        while(ss >>m)
            x.push_back(m);
    }
    sort(x.begin(),x.end());
    int t=x[0];
    for(unsigned int i=1;i<x.size();i++){
        if(t==x[i])
            b=t;
        else if(t==x[i]-2)
            a=t+1;
        t=x[i];
    }
    cout << a << ' ' << b << endl;
    return 0;
```

```
}
```

【PREV-6】翻硬币

题目

题目描述
小明正在玩一个“翻硬币”的游戏。

桌上放着排成一排的若干硬币。我们用 * 表示正面，用 o 表示反面（是小写字母，不是零

比如，可能情形是：**oo***oooo

如果同时翻转左边的两个硬币，则变为：oooo***oooo

现在小明的问题是：如果已知了初始状态和要达到的目标状态，每次只能同时翻转相邻的两个

我们约定：把翻动相邻的两个硬币叫做一步操作，那么要求：

输入格式
两行等长的字符串，分别表示初始状态和要达到的目标状态。每行的长度<1000

输出格式
一个整数，表示最小操作步数。

样例输入
****
o****o****

样例输出
5

解析

遍历 不等就翻转就行了

c++代码

```
#include <set>
#include <cstdio>
#include <cmath>
#include <cctype>
#include <cstring>
```

```cpp
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <vector>
#include <stack>
#include <queue>
#include <deque>
#include <string>
#include <algorithm>
using namespace std;
const int maxn=100000+10;

int main(){
    ios::sync_with_stdio(false);
    string s1,s2;
    cin >> s1 >> s2;
    int cnt=0;
    for(unsigned int i=0;i<s1.length()-1;i++){
        if(s1[i]!=s2[i]){
            cnt++;
            for(int j=0;j<=1;j++)
                if(s2[i+j]=='*')
                    s2[i+j]='o';
                else
                    s2[i+j]='*';
        }
    }
    cout << cnt << endl;
    return 0;
}
```