【BASIC-1】闰年判断

题目

题目描述
给定一个年份，判断这一年是不是闰年。
当以下情况之一满足时，这一年是闰年：
1. 年份是4的倍数而不是100的倍数；
2. 年份是400的倍数。
其他的年份都不是闰年。

输入格式
输入包含一个整数y，表示当前的年份。

输出格式
输出一行，如果给定的年份是闰年，则输出yes，否则输出no。

样例输入
2013

样例输出
no

解析

按照题目所描述的编写就好。
使用if语句或者像我一样用三目运算符来解决。

c++代码

```
#include <stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    printf("%s\n",(  ( n%4==0 && n%100!=0 || n%400==0 )?"yes":"
no" ) );
    return 0;
}
```

【BASIC-2】01字串

题目

题目描述
对于长度为5位的一个01串，每一位都可能是0或1，一共有32种可能。它们的前几个是：
00000
00001
00010
00011
00100
请按从小到大的顺序输出这32种01串。

输入格式
本试题没有输入。

输出格式
输出32行，按从小到大的顺序每行一个长度为5的01串。

样例输出
00000
00001
00010
00011
<以下部分省略>

解析

32次循环，每次计算二进制数值并输出就行了。
注意题目要求输出必须是5位，不足五位前导零。

c++代码

```
#include<stdio.h>
int main(){
    for(int i=0;i<32;i++){
        int a=i;
        int b[5]={0},t=0;
        while(a){
            b[t++]=a%2;
            a=a/2;
```

```
        }
        for(int j=4;j>=0;j--)
            printf("%d",b[j]);
        printf("\n");
    }
    return 0;
}
```

【BASIC-3】字母图形

题目

题目描述
利用字母可以组成一些美丽的图形，下面给出了一个例子：
ABCDEFG
BABCDEF
CBABCDE
DCBABCD
EDCBABC
这是一个5行7列的图形，请找出这个图形的规律，并输出一个n行m列的图形。

输入格式
输入一行，包含两个整数n和m，分别表示你要输出的图形的行数的列数。
1 <= n, m <= 26。

输出格式
输出n行，每个m个字符，为你的图形。

样例输入
5 7

样例输出
ABCDEFG
BABCDEF
CBABCDE
DCBABCD
EDCBABC

解析

水题，循环使用abs取绝对值来判断是输出哪个字符。

c++代码

```
#include <stdio.h>
#include <math.h>
int main(){
    int n,m;
    scanf("%d%d",&n,&m);
```

```
    for(int i=0;i<n;i++){
        for(int j=-i;j<-i+m;j++){
            char c='A'+abs(j);
            printf("%c",c);
        }
        printf("\n");;
    }
    return 0;
}
```

【BASIC-4】数列特征

题目

题目描述
给出n个数，找出这n个数的最大值，最小值，和。

输入格式
第一行为整数n，表示数的个数。
第二行有n个数，为给定的n个数，每个数的绝对值都小于10000。
1 <= n <= 10000。

输出格式
输出三行，每行一个整数。第一行表示这些数中的最大值，第二行表示这些数中的最小值，

样例输入
5
1 3 -2 4 5

样例输出
5
-2
11

解析

用三个变量记录最大值 最小值 总和 然后输出即可。

c++代码

```
#include <stdio.h>

int main(){
    int n,x;
    scanf("%d",&n);
    scanf("%d",&x);
    int maxc=x,minc=x,sum=x;
    for(int i=1;i<n;i++){
        scanf("%d",&x);
        sum+=x;
        if(x>maxc)
```

```
            maxc=x;
        if(x<minc)
            minc=x;
    }
    printf("%d\n%d\n%d\n",maxc,minc,sum);
    return 0;
}
```

【BASIC-5】查找整数

题目

题目描述
给出一个包含n个整数的数列，问整数a在数列中的第一次出现是第几个。

输入格式
第一行包含一个整数n。
第二行包含n个非负整数，为给定的数列，数列中的每个数都不大于10000。
第三行包含一个整数a，为待查找的数。
1 <= n <= 1000。

输出格式
如果a在数列中出现了，输出它第一次出现的位置(位置从1开始编号)，否则输出-1。

样例输入
6
1 9 4 8 3 9
9

样例输出
2

解析

存储数列 顺序查找就行了。

c++代码

```
#include <stdio.h>
int main(){
    int n;
    int a[1010];
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&a[i]);
    int x,cnt=-1;
    scanf("%d",&x);
    for(int i=0;i<n;i++)
        if(x==a[i]){
```

```
            cnt=i+1;
            break;
        }
    printf("%d",cnt);
    return 0;
}
```

【BASIC-6】杨辉三角形

题目

题目描述
杨辉三角形又称Pascal三角形，它的第i+1行是(a+b)i的展开式的系数。
它的一个重要性质是：三角形中的每个数字等于它两肩上的数字相加。
下面给出了杨辉三角形的前4行：

```
   1
  1 1
 1 2 1
1 3 3 1
```

给出n，输出它的前n行。

输入格式
输入包含一个数n。1 <= n <= 34。

输出格式
输出杨辉三角形的前n行。每一行从这一行的第一个数开始依次输出，中间使用一个空格分隔

样例输入

```
4
```

样例输出

```
1
1 1
1 2 1
1 3 3 1
```

解析

杨辉三角形大家都知道的。直接做一个足够大的二维数组，然后按照题目要求计算就行了。

c++代码

```
#include <stdio.h>
#include <string.h>
int main(){
    int a[50][50];
```

```
    memset(a,0,sizeof(a));
    int n;
    scanf("%d",&n);
    a[0][1]=1;
    for(int i=1;i<=n;i++){
        for(int j=1;j<=i;j++){
            a[i][j]=a[i-1][j-1]+a[i-1][j];
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

【BASIC-7】特殊的数字

题目

题目描述
153是一个非常特殊的数，它等于它的每位数字的立方和，即153=1*1*1+5*5*5+3*3*3。

输出格式
按从小到大的顺序输出满足条件的三位十进制数，每个数占一行。

解析

课后习题不解释

c++代码

```
#include <stdio.h>
int main(){
    for(int i=100;i<=999;i++){
        int a=i/100,b=(i/10)%10,c=i%10;
        if(a*a*a+b*b*b+c*c*c==i)
            printf("%d\n",i);;
    }
    return 0;
}
```

【BASIC-8】回文数

题目

题目描述
1221是一个非常特殊的数，它从左边读和从右边读是一样的，编程求所有这样的四位十进制

输出格式
按从小到大的顺序输出满足条件的四位十进制数。

解析

倒着输出就行了。

c++代码

```
#include <stdio.h>
int main(){
    for(int i=10;i<=99;i++){
        int a=i/10,b=i%10;
        printf("%d%d%d\n",i,b,a);
    }
    return 0;
}
```

【BASIC-9】特殊回文数

题目

题目描述
123321是一个非常特殊的数，它从左边读和从右边读是一样的。
输入一个正整数n， 编程求所有这样的五位和六位十进制数，满足各位数字之和等于n 。

输入格式
输入一行，包含一个正整数n。1<=n<=54。

输出格式
按从小到大的顺序输出满足条件的整数，每个整数占一行。

样例输入
52

样例输出
899998
989989
998899

解析

遍历一下就可以了。

c++代码

```
#include <stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    if(n<=45){          //当n<=45的情况下 才有5位数有可能是回文数
        for(int i=100;i<=999;i++){
            int a=i/100,b=(i%100)/10,c=i%10;
            if(2*a+2*b+c==n)
                printf("%d%d%d\n",i,b,a);
        }
    }
    for(int i=100;i<=999;i++){
        int a=i/100,b=(i%100)/10,c=i%10;
```

```
            if(2*a+2*b+2*c==n)
                printf("%d%d%d%d\n",i,c,b,a);
        }
        return 0;
    }
```

【BASIC-10】十进制转十六进制

题目

题目描述
十六进制数是在程序设计时经常要使用到的一种整数的表示方式。它有0,1,2,3,4,5
给出一个非负整数，将它表示成十六进制的形式。

输入格式
输入包含一个非负整数a，表示要转换的数。0<=a<=2147483647

输出格式
输出这个整数的16进制表示

样例输入
30

样例输出
1E

解析

循环 整除 求余 判断
我这里偷懒直接使用的库

c++代码

```
#include <iostream>
using namespace std;
int main(){
    long long a;
    cin >> a ;
    cout << uppercase <<hex << a << endl;
    return 0;
}
```

【BASIC-11】十六进制转十进制

题目

题目描述
从键盘输入一个不超过8位的正的十六进制数字符串，将它转换为正的十进制数后输出。
注：十六进制数中的10~15分别用大写的英文字母A、B、C、D、E、F表示。

样例输入
FFFF

样例输出
65535

解析

进制转换 字符处理 判断
我这里偷懒 使用的库

c++代码

```cpp
#include <iostream>
using namespace std;
int main(){
    long long a;
    cin >> hex >> a;
    cout << a << endl;
    return 0;
}
```

【BASIC-12】十六进制转八进制

题目

题目描述
给定n个十六进制正整数，输出它们对应的八进制数。

输入格式
输入的第一行为一个正整数n （1<=n<=10）。
接下来n行，每行一个由0~9、大写字母A~F组成的字符串，表示要转换的十六进制正整数，

输出格式
输出n行，每行为输入对应的八进制正整数。
【注意】
输入的十六进制数不会有前导0，比如012A。
输出的八进制数也不能有前导0。

样例输入
2
39
123ABC

样例输出
71
4435274

解析

进制转换 字符 循环
把16进制转换成2进制 然后再从二进制转换成8进制

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;
string l[16]={"0000","0001","0010","0011",
            "0100","0101","0110","0111",
            "1000","1001","1010","1011",
            "1100","1101","1110","1111"};
int main(){
    ios::sync_with_stdio(false);
    int n;
    cin >> n ;
    while(n--){
        string s;
        cin >> s;
        string s1,s2;
        for(int i=0;i<s.length();i++){
            char c=s[i];
            int a;
            if(c>='A' && c<='F')
                a=c-'A'+10;
            else
                a=c-'0';
            s1+=l[a];
        }
        int a=0,t=s1.length()%3;
        if(t==0)
            t=3;
        for(int i=0;i<s1.length();i++){
            char c=s1[i];
            a=a*2+(c-'0');
            t--;
            if(t==0){
                char cc=a+'0';
                s2+=cc;
                t=3;
                a=0;
            }
        }
        if(s2[0]=='0')
            s1.assign(s2.begin()+1,s2.end());
```

```
            else
                s1.assign(s2.begin(),s2.end());
            cout << s1 << endl;
        }
        return 0;
    }
```

【BASIC-13】数列排序

题目

题目描述
给定一个长度为n的数列，将这个数列按从小到大的顺序排列。1<=n<=200

输入格式
第一行为一个整数n。
第二行包含n个整数，为待排序的数，每个整数的绝对值小于10000。

输出格式
输出一行，按从小到大的顺序输出排序后的数列。

样例输入
5
8 3 6 4 9

样例输出
3 4 6 8 9

解析

选择排序 冒泡排序 快速排序 归并排序等等 能排序就行。

c++代码

```
#include<cstdio>
#include<algorithm>
using namespace std;

int main(){
    int n,a[205];
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&a[i]);
    sort(a,a+n);
    for(int i=0;i<n;i++){
        printf("%d",a[i]);
        if(i+1==n)
            printf("\n");
```

```
            else
                printf(" ");
        }
        return 0;
    }
```

【BASIC-14】 时间转换

题目

题目描述
给定一个以秒为单位的时间t，要求用“H:M:S”的格式来表示这个时间。H表示时间，M表示

输入格式
输入只有一行，是一个整数t（0<=t<=86399）。

输出格式
输出只有一行，是以“H:M:S”的格式所表示的时间，不包括引号。

样例输入
5436

样例输出
1:30:36

解析

取余计算。

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>
```

```cpp
using namespace std;
const int maxn=100000+10;

int main(){
    ios::sync_with_stdio(false);
    int t;
    cin >> t;
    int a=t%60,b=(t/60)%60,c=t/3600;
    cout << c << ':' << b << ':' << a << endl;
    return 0;
}
```

【BASIC-15】字符串对比

题目

题目描述
给定两个仅由大写字母或小写字母组成的字符串(长度介于1到10之间)，它们之间的关系是
1. 两个字符串长度不等。比如 Beijing 和 Hebei
2. 两个字符串不仅长度相等，而且相应位置上的字符完全一致(区分大小写)，比如 Beij
3. 两个字符串长度相等，相应位置上的字符仅在不区分大小写的前提下才能达到完全一致
4. 两个字符串长度相等，但是即使是不区分大小写也不能使这两个字符串一致。比如 Bei
编程判断输入的两个字符串之间的关系属于这四类中的哪一类，给出所属的类的编号。

输入格式
包括两行，每行都是一个字符串

输出格式
仅有一个数字，表明这两个字符串的关系编号

样例输入
```
BEIjing
beiJing
```

样例输出
```
3
```

解析

字符串对比就行，比较简单。

c++代码

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

```cpp
#include <stack>
#include <queue>
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;

int main(){
    ios::sync_with_stdio(false);
    string s1,s2;
    cin >> s1 >> s2;
    if(s1.length()!=s2.length()){
        cout << 1 << endl;
        return 0;
    }
    if(s1==s2){
        cout << 2 << endl;
        return 0;
    }
    bool flag=1;
    for(int i=0;i<s1.length() && flag;i++){
        if(s1[i]>='A' && s1[i] <='Z')
            s1[i]+=32;
        if(s2[i]>='A' && s2[i] <='Z')
            s2[i]+=32;
        if(s1[i]!=s2[i])
            flag=0;
    }
    cout << (flag ? 3:4) << endl;
    return 0;
}
```

【BASIC-16】分解质因数

题目

题目描述
求出区间[a,b]中所有整数的质因数分解。

输入格式
输入两个整数a，b。

输出格式
每行输出一个数的分解，形如k=a1*a2*a3...(a1<=a2<=a3...，k也是从小到大的)(具

样例输入
3 10

样例输出
3=3
4=2*2
5=5
6=2*3
7=7
8=2*2*2
9=3*3
10=2*5

解析

强行遍历就行了，循环。

c++代码

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

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

int main(){
    ios::sync_with_stdio(false);
    int a,b;
    cin >> a >> b;
    for(int i=a;i<=b;i++){
        int c=i;
        bool flag=0;
        cout << c << '=';
        for(int j=2;j<=c;){
            if(c%j==0){
                if(flag)
                    cout << '*';
                cout << j;
                c=c/j;
                flag=1;
                continue;
            }
            j++;
        }
        cout << endl;
    }
    return 0;
}
```

【BASIC-17】矩阵乘法

题目

题目描述
给定一个N阶矩阵A，输出A的M次幂（M是非负整数）
例如：
A =
1 2
3 4
A的2次幂
7 10
15 22

输入格式
第一行是一个正整数N、M（1<=N<=30, 0<=M<=5），表示矩阵A的阶数和要求的幂数
接下来N行，每行N个绝对值不超过10的非负整数，描述矩阵A的值

输出格式
输出共N行，每行N个整数，表示A的M次幂所对应的矩阵。相邻的数之间用一个空格隔开

样例输入
2 2
1 2
3 4

样例输出
7 10
15 22

解析

行列式的计算 好好看看线代吧~

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>
#include <functional>
using namespace std;
const int maxn=100000+10;
long long a[35][35],b[35][35],c[35][35];


void fun(int n){
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++){
            c[i][j]=0;
            for(int k=0;k<n;k++)
                c[i][j]+=a[i][k]*b[k][j];
        }
}

int main(){
    ios::sync_with_stdio(false);
    int n,m;
    cin >> n >> m;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++){
            cin >> b[i][j];
            c[i][j]=b[i][j];
        }
    for(int i=1;i<m;i++){
        for(int j=0;j<n;j++)
            for(int k=0;k<n;k++)
                a[j][k]=c[j][k];
        fun(n);
    }
    if(m==0){
        for(int i=0;i<n;i++)
            for(int j=0;j<n;j++)
                c[i][j]= i==j?1:0;
    }
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++)
```

```
            cout << c[i][j] << ' ';
        cout << endl;
    }
    return 0;
}
```

【BASIC-18】矩形面积交

题目

题目描述
平面上有两个矩形，它们的边平行于直角坐标系的X轴或Y轴。对于每个矩形，我们给出它的

输入格式
输入仅包含两行，每行描述一个矩形。
在每行中，给出矩形的一对相对顶点的坐标，每个点的坐标都用两个绝对值不超过10^7的实

输出格式
输出仅包含一个实数，为交的面积，保留到小数后两位。

样例输入
```
1 1 3 3
2 2 4 4
```

样例输出
```
1.00
```

解析

求出x轴重合长度 y轴重合长度
相乘就行了

c++代码

```cpp
#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>
```

```cpp
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;
double fun(double c[]){
    if(c[0]>c[1])
        swap(c[0],c[1]);
    if(c[2]>c[3])
        swap(c[2],c[3]);
    if(c[2]<=c[0] && c[1]<=c[3])
        return c[1]-c[0];
    else if(c[0]<=c[2] && c[3]<=c[1])
        return c[3]-c[2];
    else if(c[1]>c[2] && c[0]<c[2])
        return c[1]-c[2];
    else if(c[3]>c[0] && c[2]<c[0])
        return c[3]-c[0];
    return 0;
}
int main(){
    ios::sync_with_stdio(false);
    double x[4],y[4];
    for(int i=0;i<4;i++)
        cin >> x[i] >> y[i];
    printf("%.2lf\n",fun(x)*fun(y));
    return 0;
}
```

【BASIC-19】完美的代价

题目

题目描述

回文串，是一种特殊的字符串，它从左往右读和从右往左读是一样的。小龙龙认为回文
交换的定义是：交换两个相邻的字符
例如mamad
第一次交换 ad : mamda
第二次交换 md : madma
第三次交换 ma : madam (回文！完美！)

输入格式

第一行是一个整数N，表示接下来的字符串的长度(N <= 8000)
第二行是一个字符串，长度为N.只包含小写字母

输出格式

如果可能，输出最少的交换次数。
否则输出Impossible

样例输入

```
5
mamad
```

样例输出

```
3
```

解析

自己看吧

c++代码

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

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

int main(){
    ios::sync_with_stdio(false);
    int n,t;
    string s;
    cin >> n >> s;
    int cnt=0;
    int flag=0;
    for(int i=0;i<n/2;){
        t=-1;
        for(int j=n-i-1;j>i;j--)
            if(s[i]==s[j]){
                t=j;
                break;
            }
        if(t==-1){
            flag++;
            swap(s[i],s[i+1]);
            cnt++;
            if(flag==2)
                break;
            continue;
        }
        flag=0;
        for(int j=t;j<n-i-1;j++)
            swap(s[j],s[j+1]);
        cnt+=(n-1-i-t);
        i++;
    }
    sort(s.begin(),s.end());
    int cntt=0;
    for(int i=0;i<s.length();)
        if(i!=s.length()-1 && s[i]==s[i+1]){
            i+=2;
        }
        else if(i==s.length()-1){
```

```
            cntt++;
            i++;
        }
        else{
            cntt++;
            i++;
        }
    if(cntt>=2)
        cout << "Impossible" << endl;
    else
    cout << cnt << endl;
    return 0;
}
```

【BASIC-21】Sine之舞

题目

题目描述

最近FJ为他的奶牛们开设了数学分析课，FJ知道若要学好这门课，必须有一个好的三角
不妨设
An=sin(1−sin(2+sin(3−sin(4+...sin(n))...)
Sn=(...(A1+n)A2+n−1)A3+...+2)An+1
FJ想让奶牛们计算Sn的值，请你帮助FJ打印出Sn的完整表达式，以方便奶牛们做题。

输入格式

仅有一个数：N<201。

输出格式

请输出相应的表达式Sn，以一个换行符结束。输出中不得含有多余的空格或换行、回车符。

样例输入

3

样例输出

((sin(1)+3)sin(1−sin(2))+2)sin(1−sin(2+sin(3)))+1

解析

递归就行了。

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>
```

```cpp
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;
int N;
void An(int k,int n,bool flag){
    cout << "sin(";
    if(k!=n){
        cout << k;
        cout << (flag?'-':'+');
        An(k+1,n,!flag);
    }
    else{
        cout << k;
    }
    cout << ')';
}
void fun(int k,int n){
    if(k!=N){
        cout << '(';
        fun(k+1,n-1);
        cout << ')';
    }
    An(1,n,1);
    cout << '+' << k;
}

int main(){
    ios::sync_with_stdio(false);
    cin >> N;
    fun(1,N);
    cout << endl;
    return 0;
}
```

【BASIC-22】FJ的字符串

题目

题目描述

FJ在沙盘上写了这样一些字符串:
A1 = “A”
A2 = “ABA”
A3 = “ABACABA”
A4 = “ABACABADABACABA”
… …
你能找出其中的规律并写所有的数列AN吗?

输入格式

仅有一个数: N ≤ 26。

输出格式

请输出相应的字符串AN，以一个换行符结束。输出中不得含有多余的空格或换行、回车符。

样例输入

3

样例输出

ABACABA

解析

递归就行了。

c++代码

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

```cpp
#include <stack>
#include <queue>
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;

void fun(int n){
    if(n==0){
        cout << 'A';
        return;
    }
    fun(n-1);
    char c='A'+n;
    cout << c;
    fun(n-1);
}
int main(){
    ios::sync_with_stdio(false);
    int N;
    cin >> N;
    fun(N-1);
    cout << endl;
    return 0;
}
```

【BASIC-23】芯片测试

题目

题目描述

有n（2≤n≤20）块芯片，有好有坏，已知好芯片比坏芯片多。

每个芯片都能用来测试其他芯片。用好芯片测试其他芯片时，能正确给出被测试芯片是

给出所有芯片的测试结果，问哪些芯片是好芯片。

输入格式

输入数据第一行为一个整数n，表示芯片个数。

第二行到第n+1行为n*n的一张表，每行n个数据。表中的每个数据为0或1，在这n行中

输出格式

按从小到大的顺序输出所有好芯片的编号

样例输入

```
3
1 0 1
0 1 0
1 0 1
```

样例输出

```
1 3
```

解析

c++代码

```cpp
#include <set>
#include <cstdio>
#include <cmath>
#include <cctype>
#include <cstring>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <vector>
#include <stack>
#include <queue>
```

```cpp
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;

int c[20];
int n;
int a[20][20];
int b[20];
void fun(int k){
    if(k!=0){
        c[k-1]=1;
        fun(k-1);
        c[k-1]=0;
        fun(k-1);
    }
    else{
        bool flag=1;
        for(int i=0;i<n;i++)
            for(int j=0;j<n;j++)
                if(i!=j && c[i]==1 && a[i][j]!= (c[i]&c[j]) )
                    flag=0;
        if(flag){
            int cnt=0;
            for(int i=0;i<n;i++)
                if(c[i])
                    cnt++;
            if(cnt*2<n)
                return;
            for(int i=0;i<n;i++)
                b[i]=b[i]||c[i];
        }
    }
}

int main(){
    ios::sync_with_stdio(false);
    cin >> n;
    memset(b,0,sizeof(b));
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            cin >> a[i][j];
    fun(n);
    for(int i=0;i<n;i++)
        if(b[i])
```

```
            cout << i+1 << ' ';
        cout << endl;
        return 0;
    }
```

【BASIC-24】龟兔赛跑预测

题目

题目描述

话说这个世界上有各种各样的兔子和乌龟，但是研究发现，所有的兔子和乌龟都有一个
然而有些比赛相当漫长，全程观看会耗费大量时间，而小华发现只要在每场比赛开始后

输入格式

输入只有一行，包含用空格隔开的五个正整数v1，v2，t，s，l，其中(v1,v2<=100;t<=

输出格式

输出包含两行，第一行输出比赛结果——一个大写字母“T”或“R”或“D”，分别表示乌龟获胜，
第二行输出一个正整数，表示获胜者（或者双方同时）到达终点所耗费的时间（秒数）。

样例输入

```
10 5 5 2 20
```

样例输出

```
D
4
```

解析

c++代码

```cpp
#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>
```

```cpp
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;

int main(){
    ios::sync_with_stdio(false);
    int v1,v2,t,s,l;
    cin >> v1 >> v2 >> t >> s >> l;
    int x=0,l1=0,l2=0,tt=0;
    while(1){
        x++;
        if(tt>0){
            tt--;
        }
        else{
            l1+=v1;
        }
        l2+=v2;
        if(l2==l1 && l2==l){
            cout <<'D' << endl << x << endl;break;}
        else if(l2==l){
            cout << "T" << endl << x << endl;break;}
        else if(l1==l){
            cout << 'R' << endl << x << endl;break;}
        if(l1-l2>=t && tt==0)
            tt=s;
    }
    return 0;
}
```

【BASIC-25】回形取数

题目

题目描述
回形取数就是沿矩阵的边取数，若当前方向上无数可取或已经取过，则左转90度。一开始位

输入格式
输入第一行是两个不超过200的正整数m, n，表示矩阵的行和列。接下来m行每行n个整数，

输出格式
输出只有一行，共mn个数，为输入矩阵回形取数得到的结果。数之间用一个空格分隔，行末

样例输入
```
3 3
1 2 3
4 5 6
7 8 9
```

样例输出
```
1 4 7 8 9 6 3 2 5
```

解析

水题

c++代码

```cpp
#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>
```

```cpp
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;
int l[4][2]={1,0,0,1,-1,0,0,-1};
int main(){
    ios::sync_with_stdio(false);
    int m,n;
    int a[210][210],b[210][210];
    memset(b,0,sizeof(b));
    cin >> m >> n;
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            cin >> a[i][j] ;
    int x=0,y=0,k=0;
    for(int i=0;i<m*n;i++){
        cout << a[x][y];
        if(i!=m*n-1)
            cout << ' ';
        b[x][y]=1;
        int newx=x+l[k][0],newy=y+l[k][1];
        if(b[newx][newy] || newx==m || newy==n || newy==-1 || n
ewx==-1){
            k=(k+1)%4;
            newx=x+l[k][0],newy=y+l[k][1];
        }
        x=newx;
        y=newy;
    }
    cout << endl;
    return 0;
}
```

【BASIC-26】报时助手

题目

题目描述

给定当前的时间，请用英文的读法将它读出来。
时间用时h和分m表示，在英文的读法中，读一个时间的方法是：
如果m为0，则将时读出来，然后加上“o'clock”，如3:00读作“three o'clock”。
如果m不为0，则将时读出来，然后将分读出来，如5:30读作“five thirty”。
时和分的读法使用的是英文数字的读法，其中0~20读作：
0:zero, 1: one, 2:two, 3:three, 4:four, 5:five, 6:six, 7:seven,
30读作thirty，40读作forty，50读作fifty。
对于大于20小于60的数字，首先读整十的数，然后再加上个位数。如31首先读30再加1
按上面的规则21:54读作“twenty one fifty four”，9:07读作“nine seven”，

输入格式

输入包含两个非负整数h和m，表示时间的时和分。非零的数字前没有前导0。h小于24，m小

输出格式

输出时间时刻的英文。

样例输入

```
0 15
```

样例输出

```
zero fifteen
```

解析

c++代码

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

```cpp
#include <stack>
#include <queue>
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=100000+10;
string s[61]={"zero","one","two","three","four","five","six","seven","eight","nine","ten","eleven","twelve","thirteen","fourteen","fifteen","sixteen","seventeen","eighteen","nineteen","twenty"};
int main(){
    s[30]="thirty";
    s[40]="forty";
    s[50]="fifty";
    ios::sync_with_stdio(false);
    int h,m;
    cin >> h >> m;
    if(h<=20)
        cout << s[h] << ' ';
    else
        cout << s[20] << ' '<< s[h%20] << ' ';
    if(m==0)
        cout << "o'clock" << endl;
    else if(m>20 && m!=30 &&m!=40 && m!=50){
        int a=m/10,b=m%10;
        cout << s[a*10] << ' ' << s[b] << endl;
    }
    else
        cout << s[m] << endl;
    return 0;
}
```

【BASIC-27】2n皇后问题

题目

题目描述
给定一个n*n的棋盘，棋盘中有一些位置不能放皇后。现在要向棋盘中放入n个黑皇后和n个

输入格式
输入的第一行为一个整数n，表示棋盘的大小。
　　接下来n行，每行n个0或1的整数，如果一个整数为1，表示对应的位置可以放皇后，如

输出格式
输出一个整数，表示总共有多少种放法。

样例输入
```
4
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
```

样例输出
```
2
```

解析

回溯法解决。

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>
```

```cpp
#include <queue>
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
int v[10][10];
int a[15],b[15];
int n;
int funw(){
    int cnt=0,T=2;
    memset(b,0,sizeof(b));
    for(int i=0;i<n;i++)
        if(v[0][i] && a[1]!=i+1 ){
            b[1]=i+1;
            break;
        }
    while(b[1]){
        if(T==n+1){
            T--;
            cnt++;
        }
        int c=b[T];
        b[T]=0;
        for(int i=c;i<n;i++)
            if(v[T-1][i] && a[T]!=i+1){
                b[T]=i+1;
                break;
            }
        if(!b[T]){
            T--;
            continue;
        }
        bool flag=1;
        for(int i=1;i<T && flag ; i++)
            if(b[i]==b[T] || T-b[T]==i-b[i] || T+b[T]==i+b[i])
                flag=0;
        if(flag)
            T++;
    }
    return cnt;
}
int funb(){
    int cnt=0,T=2;
    memset(a,0,sizeof(a));
    for(int i=0;i<n;i++)
        if(v[0][i]){
```

```
            a[1]=i+1;
            break;
        }
    while(a[1]){
        if(T==n+1){
            T--;
            cnt+=funw();
            continue;
        }
        int c=a[T];
        a[T]=0;
        for(int i=c;i<n;i++)
            if(v[T-1][i]){
                a[T]=i+1;
                break;
            }
        if(!a[T]){
            T--;
            continue;
        }
        bool flag=1;
        for(int i=1;i<T && flag;i++)
            if(a[i]==a[T]|| T-a[T]==i-a[i] || T+a[T]==i+a[i])
                flag=0;
        if(flag)
            T++;
    }
    return cnt;
}
int main(){
    ios::sync_with_stdio(false);
    cin >> n;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            cin >> v[i][j];
    cout << funb() << endl;
    return 0;
}
```

【BASIC-28】Huffuman树

题目

题目描述

Huffman树在编码中有着广泛的应用。在这里，我们只关心Huffman树的构造过程。
给出一列数{pi}={p0, p1, …, pn-1}，用这列数构造Huffman树的过程如下：
1. 找到{pi}中最小的两个数，设为pa和pb，将pa和pb从{pi}中删除掉，然后将它们
2. 重复步骤1，直到{pi}中只剩下一个数。
在上面的操作过程中，把所有的费用相加，就得到了构造Huffman树的总费用。
本题任务：对于给定的一个数列，现在请你求出用该数列构造Huffman树的总费用。

例如，对于数列{pi}={5, 3, 8, 2, 9}，Huffman树的构造过程如下：
1. 找到{5, 3, 8, 2, 9}中最小的两个数，分别是2和3，从{pi}中删除它们并将和
2. 找到{5, 8, 9, 5}中最小的两个数，分别是5和5，从{pi}中删除它们并将和10加
3. 找到{8, 9, 10}中最小的两个数，分别是8和9，从{pi}中删除它们并将和17加入
4. 找到{10, 17}中最小的两个数，分别是10和17，从{pi}中删除它们并将和27加入
5. 现在，数列中只剩下一个数27，构造过程结束，总费用为5+10+17+27=59。

输入格式

输入的第一行包含一个正整数n（n<=100）。
接下来是n个正整数，表示p0, p1, …, pn-1，每个数不超过1000。

输出格式

输出用这些数构造Huffman树的总费用。

样例输入

```
5
5 3 8 2 9
```

样例输出

```
59
```

解析

c++代码

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

```cpp
#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=100000+10;
bool cmp(int a,int b){
    return a>b;
}
int main(){
    ios::sync_with_stdio(false);
    int n,sum=0;
    int a[110];
    cin >> n;
    for(int i=0;i<n;i++)
        cin >> a[i];
    sort(a,a+n,cmp);
    for(int i=n-1;i>0;i--){
        sum+=a[i]+a[i-1];
        a[i-1]=a[i]+a[i-1];
        sort(a,a+i,cmp);
    }
    cout << sum << endl;
    return 0;
}
```

【BASIC-29】高精度加法

题目

题目描述
输入两个整数a和b，输出这两个整数的和。a和b都不超过100位。

算法描述
由于a和b都比较大，所以不能直接使用语言中的标准数据类型来存储。对于这种问题，
定义一个数组A，A[0]用于存储a的个位，A[1]用于存储a的十位，依此类推。同样可以
计算c = a + b的时候，首先将A[0]与B[0]相加，如果有进位产生，则把进位（即和
最后将C输出即可。

输入格式
输入包括两行，第一行为一个非负整数a，第二行为一个非负整数b。两个整数都不超过100

输出格式
输出一行，表示a + b的值。

样例输入
```
20100122201001221234567890
2010012220100122
```

样例输出
```
20100122203011233454668012
```

解析

高精度加法，模拟一下就行了。

c++代码

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

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

int main(){
    ios::sync_with_stdio(false);
    int a[110],b[110],c[110];
    memset(a,0,sizeof(a));
    memset(b,0,sizeof(b));
    memset(c,0,sizeof(c));
    string s1,s2;
    cin >> s1 >> s2;
    for(int i=s1.length()-1,j=0;i>=0;i--,j++)
        a[i]=s1[j]-'0';
    for(int i=s2.length()-1,j=0;i>=0;i--,j++)
        b[i]=s2[j]-'0';
    int len=max(s1.length(),s2.length());
    for(int i=0;i<=len;i++){
        c[i]=a[i]+b[i]+c[i];
        c[i+1]=c[i]/10;
        c[i]=c[i]%10;
    }
    if(c[len])
        cout << c[len];
    for(int i=len-1;i>=0;i--)
        cout << c[i];
    cout << endl;
    return 0;
}
```

【BASIC-30】阶乘计算

题目

题目描述
输入一个正整数n，输出n!的值。
其中n!=1*2*3*…*n。

算法描述
n!可能很大，而计算机能表示的整数范围有限，需要使用高精度计算的方法。使用一个
将a乘以一个整数k变为将数组A的每一个元素都乘以k，请注意处理相应的进位。
首先将a设为1，然后乘2，乘3，当乘到n时，即得到了n!的值。

输入格式
输入包含一个正整数n，n<=1000。

输出格式
输出n!的准确值。

样例输入
10

样例输出
3628800

解析

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>
```

```cpp
#include <deque>
#include <string>
#include <algorithm>
#include <functional>
using namespace std;
const int maxn=5000+10;

int main(){
    ios::sync_with_stdio(false);
    int a[maxn],b[maxn];
    memset(a,0,sizeof(a));
    memset(b,0,sizeof(b));
    int n,len=1;
    cin >> n;
    a[0]=1;
    for(int i=2;i<=n;i++){
        for(int j=0;j<len+4;j++)
            b[j]=a[j]*i;
        for(int j=0;j<len|| (j>=len && b[j]);j++){
            a[j]=b[j]%10;
            b[j+1]+=b[j]/10;
            if(j>=len)
                len=j+1;
        }
    }
    for(int i=len-1;i>=0;i--)
        cout << a[i];
    cout << endl;
    return 0;
}
```