***油田问题（深搜）***

**#include<stdio.h>**

**#include "string.h"**

**using namespace std;**

**char ch[107][107];**

**int vis[105][105];**

**int dir[8][2]={1,0,0,-1,-1,0,0,1,1,1,1,-1,-1,-1,-1,1};**

**int n,m;**

**bool check(int x,int y)**

**{**

**if(x>=0&&y>=0&&x<n&&y<m&&ch[x][y]=='@')return true;**

**return false;**

**}**

**void dfs(int x,int y)**

**{**

**for(int i=0;i<8;i++)**

**{**

**int xx=dir[i][0]+x;**

**int yy=dir[i][1]+y;**

**if(check(xx, yy)&&!vis[xx][yy])**

**{**

**vis[xx][yy]=1;**

**dfs(xx,yy);**

**}**

**}**

**}**

**int main()**

**{**

**while(scanf("%d%d",&n,&m)&&(m||n))**

**{**

**for(int i=0;i<n;i++)scanf("%s",ch[i]);**

**memset(vis, 0, sizeof(vis));**

**int sum=0;**

**for(int i=0;i<n;i++)**

**{**

**for(int j=0;j<m;j++)if(!vis[i][j]&&ch[i][j]=='@') {sum++;dfs(i,j);}**

**}**

**printf("%d\n",sum);**

**}**

**return 0;**

**}**

**红黑块问题（广搜**

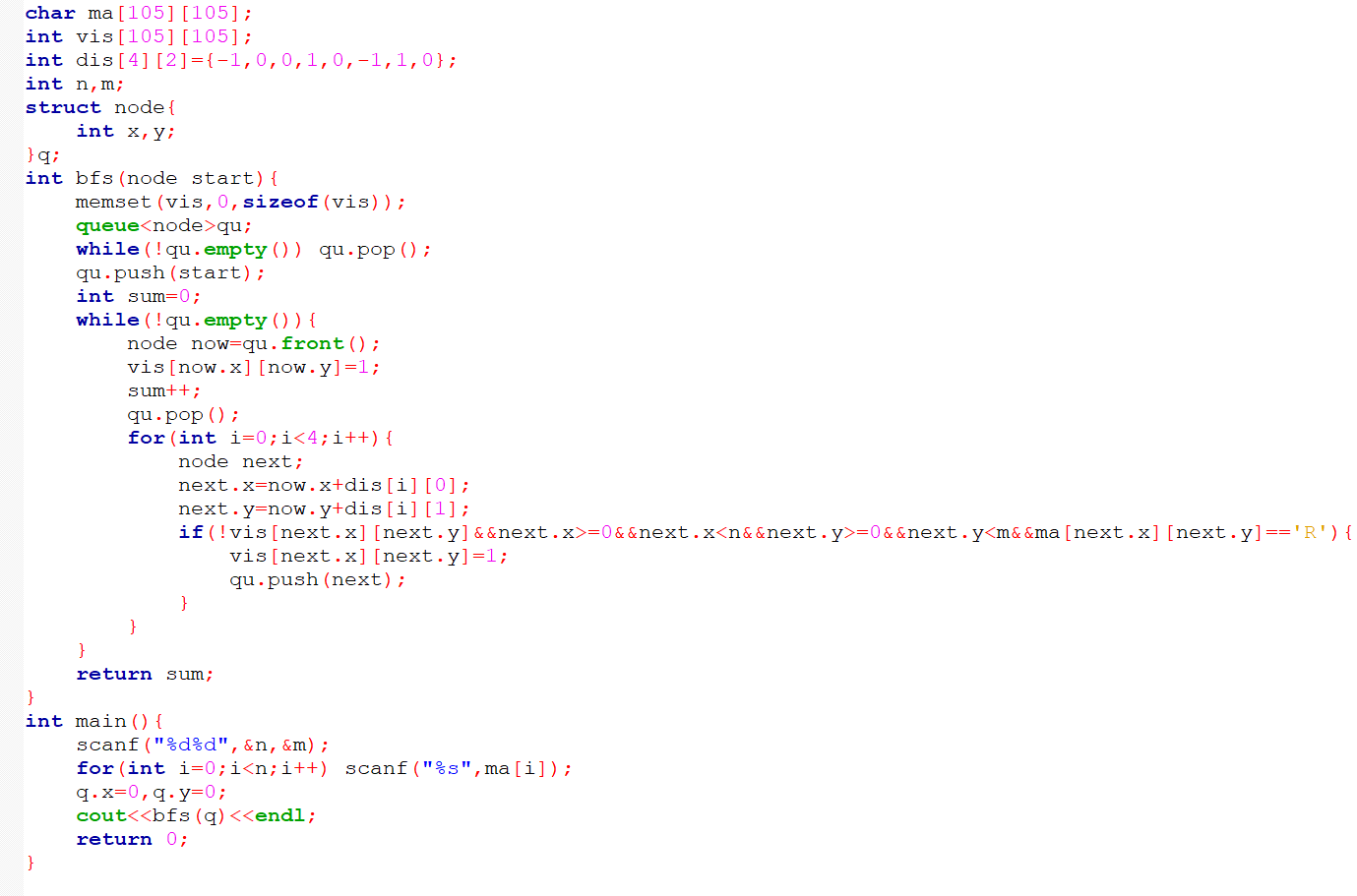
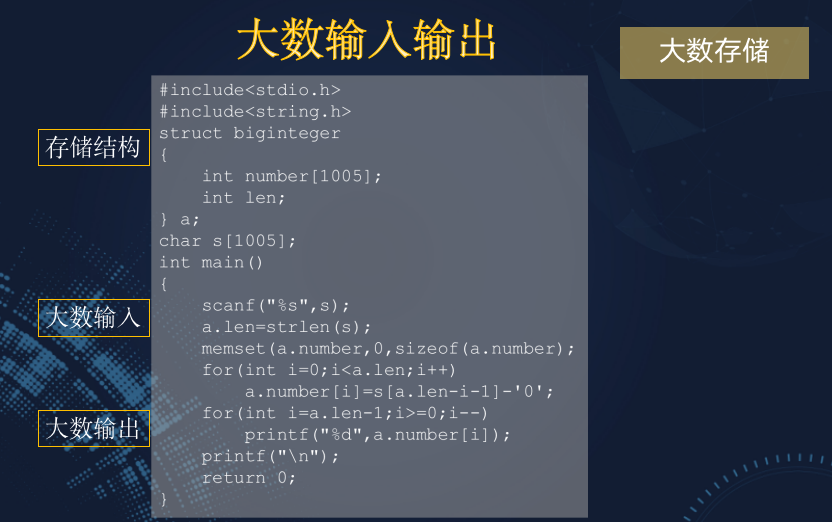
**本道题目其实就是让大家求一个从起点开始的4个方向上能连通的R的数量**

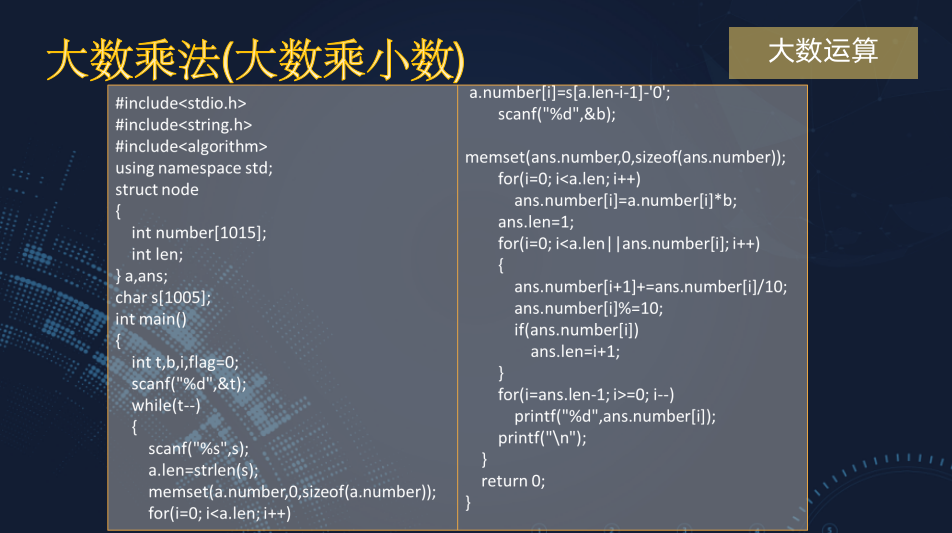
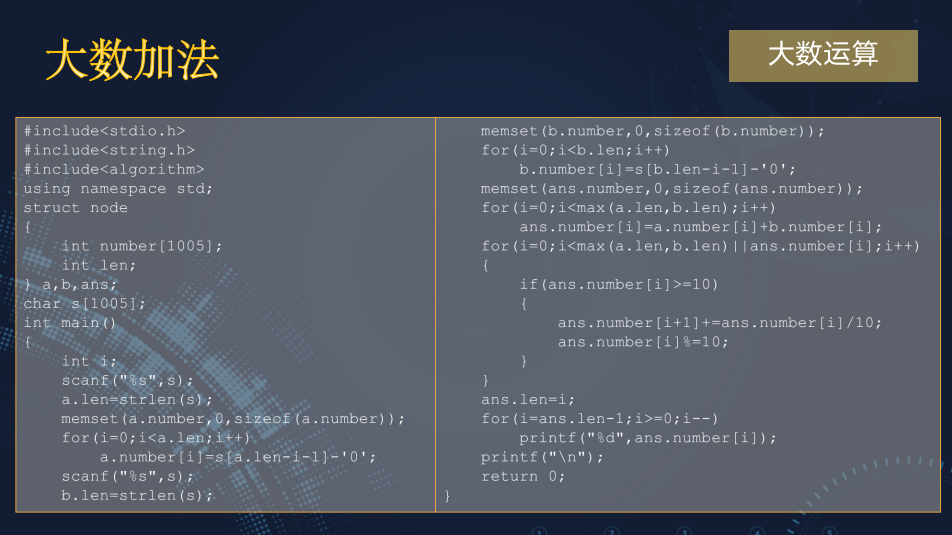
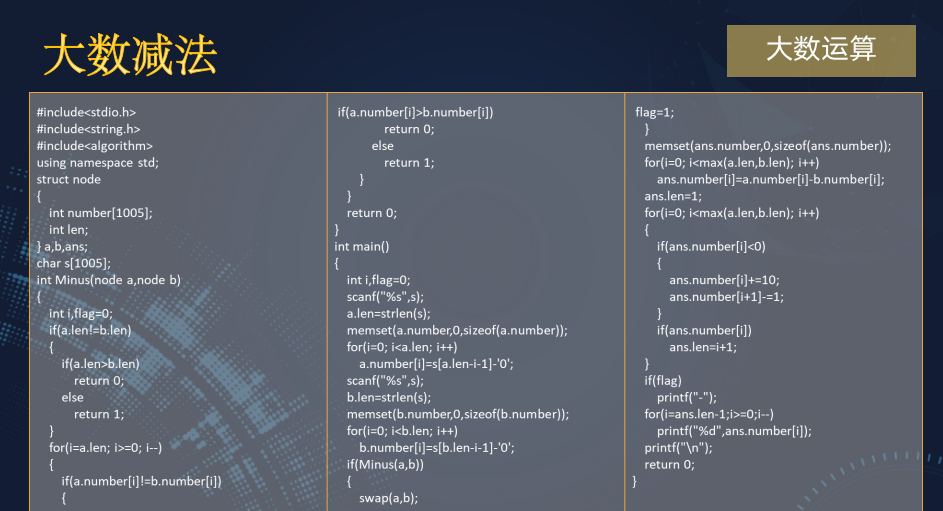
**可以考虑用bfs，先将起点左上角的元素放入队列中**

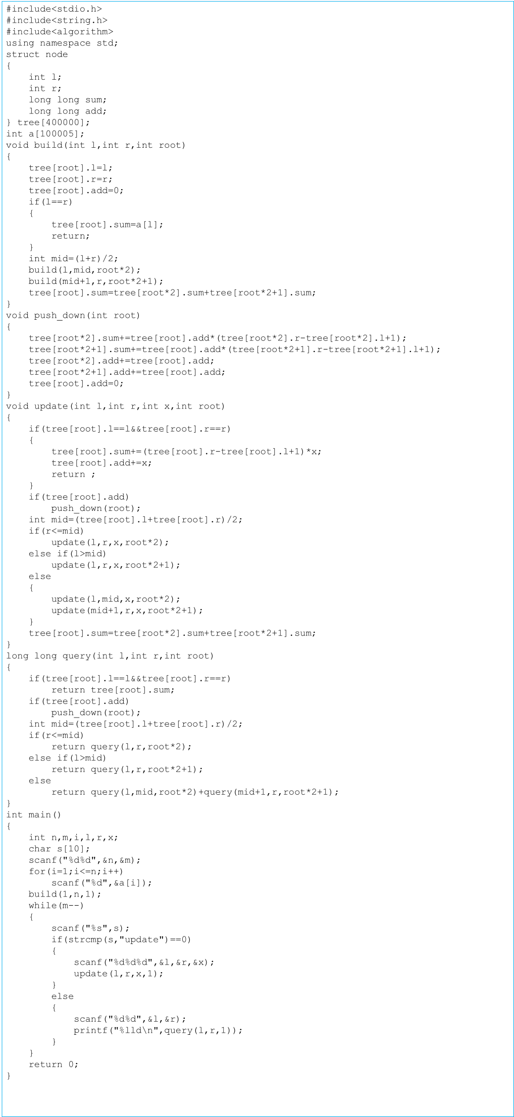
**起点放入队列中后，去看他的4个方向是否符合要求**

**既要在边界内，也要是R，将所有符合的点放入队列中**

**依次去处理，直到队列为空结束**





****

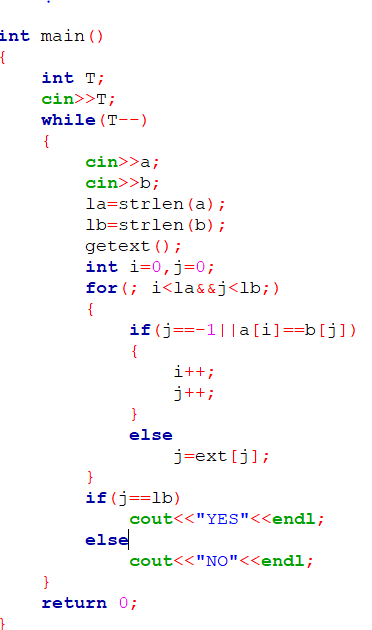
# 树

# 多源最短路径

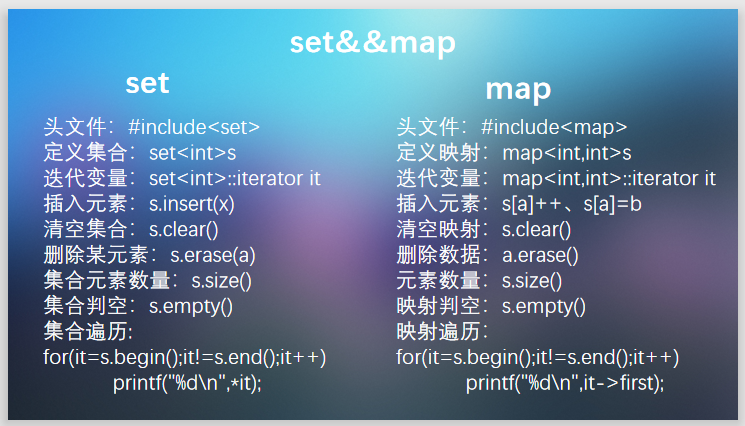
# Kmp

字符串匹配问题：

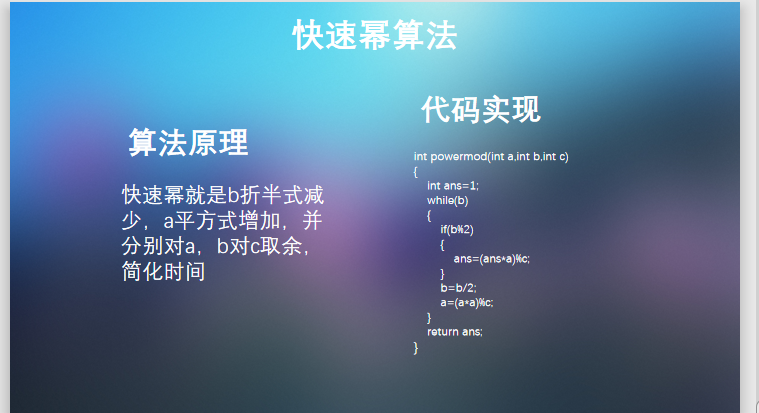
给出两个序列，问第二个串是否在第一个串内







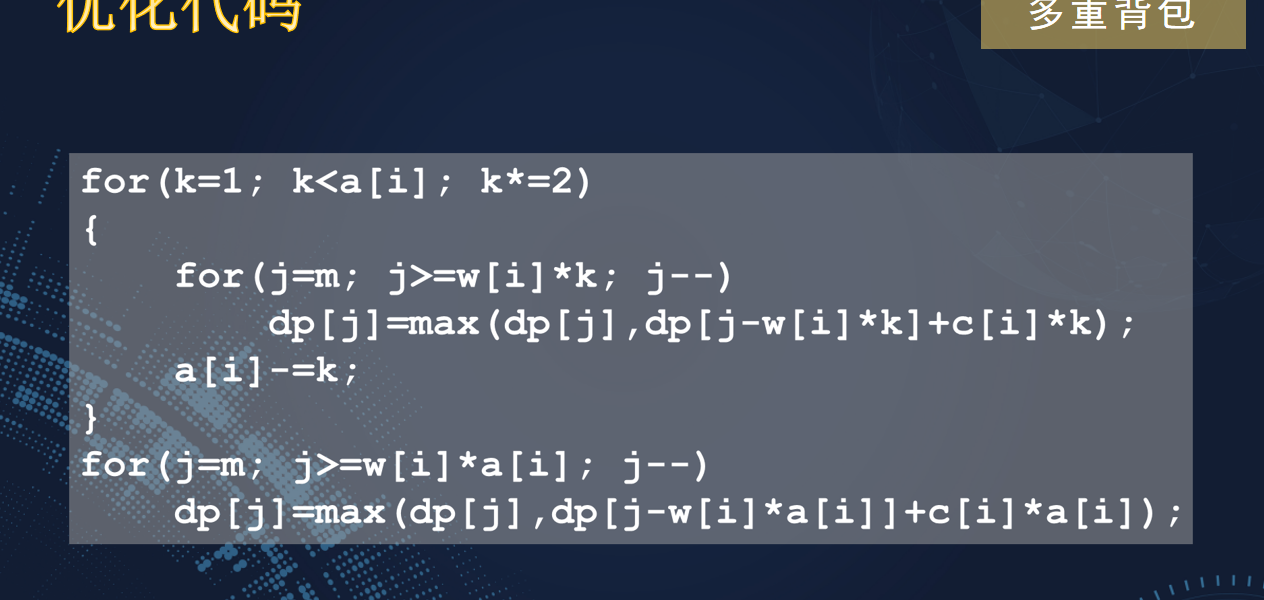












（一）**巴什博奕（Bash Game）：**只有一堆n个物品，两个人轮流从这堆物品中取物，规  
定每次至少取一个，最多取m个。最后取光者得胜。

    显然，如果n=m+1，那么由于一次最多只能取m个，所以，无论先取者拿走多少个，后取者都能够一次拿走剩余的物品，后者取胜。因此我们发现了如何取胜的法则：如果n=（m+1)\*r+s，（r为任意自然数，s≤m),那么先取者要拿走s个物品，如果后取者拿走k（≤m)个，那么先取者再拿走m+1-k个，结果剩下（m+1）（r-1）个，以后保持这样的取法，那么先取者肯定获胜。总之，要保持给对手留下（m+1）的倍数，就能最后获胜。  
    这个游戏还可以有一种变相的玩法：两个人轮流报数，每次至少报一个，最多报十  
个，谁能报到100者胜。

下面介绍分析此类题目的通用方法：P/N分析：

P点： 即必败点，某玩家位于此点，只要对方无失误，则必败；

N点： 即必胜点，某玩家位于此点，只要自己无失误，则必胜。

三个定理：

定理：

     一、 所有终结点都是必败点P（上游戏中，轮到谁拿牌，还剩0张牌的时候，此人就输了，因为无牌可取）；

    二、所有一步能走到必败点P的就是N点；

    三、通过一步操作只能到N点的就是P点；

       首先明确使用巴什博弈有个前提（特点）每次拿的数量是从1-m的不间断整数开始的。注意这一点很重要。无此条件巴什博弈不成立！

**Problem Description**

虽然不想，但是现实总归是现实，Lele始终没有逃过退学的命运，因为他没有拿到奖学金。现在等待他的，就是像FarmJohn一样的农田生涯。  
  
要种田得有田才行，Lele听说街上正在举行一场别开生面的拍卖会，拍卖的物品正好就是一块20亩的田地。于是，Lele带上他的全部积蓄，冲往拍卖会。  
  
后来发现，整个拍卖会只有Lele和他的死对头Yueyue。  
  
通过打听，Lele知道这场拍卖的规则是这样的：刚开始底价为0，两个人轮流开始加价，不过每次加价的幅度要在1～N之间，当价格大于或等于田地的成本价 M 时，主办方就把这块田地卖给这次叫价的人。  
  
Lele和Yueyue虽然考试不行，但是对拍卖却十分精通，而且他们两个人都十分想得到这块田地。所以他们每次都是选对自己最有利的方式进行加价。  
  
由于Lele字典序比Yueyue靠前，所以每次都是由Lele先开始加价，请问，第一次加价的时候，  
Lele要出多少才能保证自己买得到这块地呢？

**Input**

本题目包含多组测试，请处理到文件结束(EOF)。每组测试占一行。  
每组测试包含两个整数M和N(含义见题目描述，0<N，M<1100)

**Output**

对于每组数据，在一行里按递增的顺序输出Lele第一次可以加的价。两个数据之间用空格隔开。  
如果Lele在第一次无论如何出价都无法买到这块土地，就输出"none"。

1. #include<cstdio>
2. #include<algorithm>
3. **using** **namespace** std;
5. int main()
6. {
7. freopen("game.in","r",stdin); freopen("game.out","w",stdout);
8. int sum,maxnum;
9. **while**(scanf("%d%d",&sum,&maxnum)!=EOF)
10. {
11. **if**(sum<=maxnum)
12. {
13. **for**(int i=sum;i<=maxnum;i++) printf(i==sum?"%d":" %d",i);
14. printf("/n");
15. }
16. **else** **if**(sum%(maxnum+1)!=0)
17. {
18. int result=sum%(maxnum+1);
19. printf("%d/n",result);
20. }
21. **else** printf("none/n");
22. }
23. **return** 0;
24. }

**（二）威佐夫博奕（Wythoff Game）：**有两堆各若干个物品，两个人轮流从某一堆或同  
时从两堆中取同样多的物品，规定每次至少取一个，多者不限，最后取光者得胜。  
  
    这种情况下是颇为复杂的。我们用（ak，bk）（ak ≤ bk ,k=0，1，2，…,n)表示两堆物品的数量并称其为**局势**，如果甲面对（0，0），那么甲已经输了，这种局势我们称为**奇异局势**。前几个奇异局势是：（0，0）、（1，2）、（3，5）、（4，7）、（6，10）、（8，13）、（9，15）、（11，18）、（12，20）。

    可以看出,a0=b0=0,ak是未在前面出现过的最小自然数,而 bk= ak + k，奇异局势有  
如下三条性质：

    1。任何自然数都包含在一个且仅有一个奇异局势中。  
    由于ak是未在前面出现过的最小自然数，所以有ak > ak-1 ，而 bk= ak + k > ak-1 + k-1 = bk-1 > ak-1 。所以性质1。成立。  
    2。任意操作都可将奇异局势变为非奇异局势。  
    事实上，若只改变奇异局势（ak，bk）的某一个分量，那么另一个分量不可能在其他奇异局势中，所以必然是非奇异局势。如果使（ak，bk）的两个分量同时减少，则由于其差不变，且不可能是其他奇异局势的差，因此也是非奇异局势。  
    3。采用适当的方法，可以将非奇异局势变为奇异局势。

    假设面对的局势是（a,b），若 b = a，则同时从两堆中取走 a 个物体，就变为了奇异局势（0，0）；如果a = ak ，b > bk，那么，取走b  – bk个物体，即变为奇异局势；如果 a = ak ，  b < bk ,则同时从两堆中拿走 ak – （ab – ak）个物体,变为奇异局势（ ab – ak , ab – ak+ b – ak）；如果a > ak ，b= ak + k,则从第一堆中拿走多余的数量a – ak 即可；如果a < ak ，b= ak + k,分两种情况，第一种，a=aj （j < k）,从第二堆里面拿走 b – bj 即可；第二种，a=bj （j < k）,从第二堆里面拿走 b – aj 即可。

    从如上性质可知，两个人如果都采用正确操作，那么面对非奇异局势，先拿者必胜；反之，则后拿者取胜。

    那么任给一个局势（a，b），怎样判断它是不是奇异局势呢？我们有如下公式：

    ak =[k（1+√5）/2]，bk= ak + k  （k=0，1，2，…,n 方括号表示取整函数)  
奇妙的是其中出现了黄金分割数（1+√5）/2 = 1。618…,因此,由ak，bk组成的矩形近似为黄金矩形，由于2/（1+√5）=（√5-1）/2，可以先求出j=[a（√5-1）/2]，若a=[（1+√5）/2]，那么a = aj，bj = aj + j，若不等于，那么a = aj+1，bj+1 = aj+1+ j + 1，若都不是，那么就不是奇异局势。然后再按照上述法则进行，一定会遇到奇异局势。

（三）**尼姆博奕（Nimm Game）：**有三堆各若干个物品，两个人轮流从某一堆取任意多的  
物品，规定每次至少取一个，多者不限，最后取光者得胜。

    这种情况最有意思，它与二进制有密切关系，我们用（a，b，c）表示某种局势，首先（0，0，0）显然是奇异局势，无论谁面对奇异局势，都必然失败。第二种奇异局势是（0，n，n），只要与对手拿走一样多的物品，最后都将导致（0，0，0）。仔细分析一下，（1，2，3）也是奇异局势，无论对手如何拿，接下来都可以变为（0，n，n）的情形。

    计算机算法里面有一种叫做按位模2加，也叫做异或的运算，我们用符号（+）表示这种运算。这种运算和一般加法不同的一点是1+1=0。先看（1，2，3）的按位模2加的结果：

1 =二进制01  
2 =二进制10  
3 =二进制11 （+）  
———————  
0 =二进制00 （注意不进位）

    对于奇异局势（0，n，n）也一样，结果也是0。

    任何奇异局势（a，b，c）都有a（+）b（+）c =0。

如果我们面对的是一个非奇异局势（a，b，c），要如何变为奇异局势呢？假设 a < b< c,我们只要将 c 变为 a（+）b,即可,因为有如下的运算结果: a（+）b（+）(a（+）b)=(a（+）a)（+）(b（+）b)=0（+）0=0。要将c 变为a（+）b，只要从 c中减去 c-（a（+）b）即可。  
  
    例1。（14，21，39），14（+）21=27，39-27=12，所以从39中拿走12个物体即可达到奇异局势（14，21，27）。

    例2。（55，81，121），55（+）81=102，121-102=19，所以从121中拿走19个物品就形成了奇异局势（55，81，102）。

    例3。（29，45，58），29（+）45=48，58-48=10，从58中拿走10个，变为（29，45，48）。

**Problem Description**

一年在外 父母时刻牵挂  
春节回家 你能做几天好孩子吗  
寒假里尝试做做下面的事情吧  
  
陪妈妈逛一次菜场  
悄悄给爸爸买个小礼物  
主动地 强烈地 要求洗一次碗  
某一天早起 给爸妈用心地做回早餐  
  
如果愿意 你还可以和爸妈说  
咱们玩个小游戏吧 ACM课上学的呢～  
  
下面是一个二人小游戏：桌子上有M堆扑克牌；每堆牌的数量分别为Ni(i=1…M)；两人轮流进行；每走一步可以任意选择一堆并取走其中的任意张牌；桌子上的扑克全部取光，则游戏结束；最后一次取牌的人为胜者。  
现在我们不想研究到底先手为胜还是为负，我只想问大家：  
——“先手的人如果想赢，第一步有几种选择呢？”

**Input**

输入数据包含多个测试用例，每个测试用例占2行，首先一行包含一个整数M(1<M<=100)，表示扑克牌的堆数，紧接着一行包含M个整数Ni(1<=Ni<=1000000，i=1…M)，分别表示M堆扑克的数量。M为0则表示输入数据的结束。

1. //运用了重要性质：a^a=0
2. #include<cstdio>
3. #include<algorithm>
4. **using** **namespace** std;
5. #define N 100+10
7. int main()
8. {
9. freopen("game.in","r",stdin); freopen("game.out","w",stdout);
10. int heapnum,heap[N];
11. **while**(scanf("%d",&heapnum)!=EOF && heapnum)
12. {
13. int sum=0,ans=0;
14. **for**(int i=1;i<=heapnum;i++)
15. {
16. scanf("%d",&heap[i]); sum^=heap[i];
17. }
18. **for**(int i=1;i<=heapnum;i++)
19. {
20. **if**(heap[i]>(sum^heap[i])) ans++;//大于号的优先级要高于异或运算！
21. }
22. printf("%d/n",ans);
23. }
24. **return** 0;
25. }