

第十届全国青少年信息学奥林匹克联赛复赛试题

<http://www.oifans.cn>

（普及组 三小时完成）

不高兴的津津

(unhappy.pas/dpr/c/cpp)

【问题描述】

津津上初中了。妈妈认为津津应该更加用功学习，所以津津除了上学之外，还要参加妈妈为她报名的各科复习班。另外每周妈妈还会送她去学习朗诵、舞蹈和钢琴。但是津津如果一天上课超过八个小时就会不高兴，而且上得越久就会越不高兴。假设津津不会因为其它事不高兴，并且她的不高兴不会持续到第二天。请你帮忙检查一下津津下周的日程安排，看看下周她会不会不高兴；如果会的话，哪天最不高兴。

【输入文件】

输入文件 `unhappy.in` 包括七行数据，分别表示周一到周日的日程安排。每行包括两个小于 10 的非负整数，用空格隔开，分别表示津津在学校上课的时间和妈妈安排她上课的时间。

【输出文件】

输出文件 `unhappy.out` 包括一行，这一行只包含一个数字。如果不会不高兴则输出 0，如果会则输出最不高兴的是周几（用 1, 2, 3, 4, 5, 6, 7 分别表示周一，周二，周三，周四，周五，周六，周日）。如果有两天或两天以上不高兴的程度相当，则输出时间最靠前的一天。

【样例输入】

5 3

6 2

7 2

5 3

5 4

0 4

0 6

【样例输出】

花生采摘

(peanuts.pas/dpr/c/cpp)

【问题描述】

鲁宾逊先生有一只宠物猴，名叫多多。这天，他们两个正沿着乡间小路散步，突然发现路边的告示牌上贴着一张小小的纸条：“欢迎免费品尝我种的花生！——熊字”。

鲁宾逊先生和多多都很开心，因为花生正是他们的最爱。在告示牌背后，路边真的有一块花生田，花生植株整齐地排列成矩形网格（如图 1）。有经验的多多一眼就能看出，每棵花生植株下的花生有多少。为了训练多多的算术，鲁宾逊先生说：“你先找出花生最多的植株，去采摘它的花生；然后再找出剩下的植株里花生最多的，去采摘它的花生；依此类推，不过你一定要在我限定的时间内回到路边。”

我们假定多多在每个单位时间内，可以做下列四件事情中的一件：

- 1) 从路边跳到最靠近路边（即第一行）的某棵花生植株；
- 2) 从一棵植株跳到前后左右与之相邻的另一棵植株；
- 3) 采摘一棵植株下的花生；
- 4) 从最靠近路边（即第一行）的某棵花生植株跳回路边。

现在给定一块花生田的大小和花生的分布，请问在限定时间内，多多最多可以采到多少个花生？注意可能只有部分植株下面长有花生，假设这些植株下的花生个数各不相同。

例如在图 2 所示的花生田里，只有位于(2, 5), (3, 7), (4, 2), (5, 4)的植株下长有花生，个数分别为 13, 7, 15, 9。沿着图示的路线，多多在 21 个单位时间内，最多可以采到 37 个花生。

【输入文件】

输入文件 peanuts.in 的第一行包括三个整数，M, N 和 K，用空格隔开；表示花生田的大小为 $M * N$ ($1 \leq M, N \leq 20$)，多多采花生的限定时间为 K ($0 \leq K \leq 1000$) 个单位时间。接下来的 M 行，每行包括 N 个非负整数，也用空格隔开；第 $i + 1$ 行的第 j 个整数 P_{ij} ($0 \leq P_{ij} \leq 500$) 表示花生田里植株(i, j)下花生的数目，0 表示该植株下没有花生。

【输出文件】

输出文件 peanuts.out 包括一行，这一行只包含一个整数，即在限定时间内，多多最多可以采到花生的个数。

【样例输入 1】

6 7 21
0 0 0 0 0 0 0
0 0 0 0 13 0 0
0 0 0 0 0 0 7
0 15 0 0 0 0 0
0 0 0 9 0 0 0
0 0 0 0 0 0 0

【样例输出 1】

37

【样例输入 2】

6 7 20
0 0 0 0 0 0 0
0 0 0 0 13 0 0
0 0 0 0 0 0 7
0 15 0 0 0 0 0
0 0 0 9 0 0 0
0 0 0 0 0 0 0

【样例输出 2】

28

FBI 树

(fbi.pas/dpr/c/cpp)

【问题描述】

我们可以把由“0”和“1”组成的字符串分为三类：全“0”串称为B串，全“1”串称为I串，既含“0”又含“1”的串则称为F串。

FBI树是一种二叉树[1]，它的结点类型也包括F结点，B结点和I结点三种。由一个长度为 $2N$ 的“01”串 S 可以构造出一棵FBI树 T ，递归的构造方法如下：

- 1) T 的根结点为 R ，其类型与串 S 的类型相同；
- 2) 若串 S 的长度大于1，将串 S 从中间分开，分为等长的左右子串 S_1 和 S_2 ；由左子串 S_1 构造 R 的左子树 T_1 ，由右子串 S_2 构造 R 的右子树 T_2 。

现在给定一个长度为 $2N$ 的“01”串，请用上述构造方法构造出一棵FBI树，并输出它的后序遍历[2]序列。

【输入文件】

输入文件 fbi.in 的第一行是一个整数 N ($0 \leq N \leq 10$)，第二行是一个长度为 $2N$ 的“01”串。

【输出文件】

输出文件 fbi.out 包括一行，这一行只包含一个字符串，即 FBI 树的后序遍历序列。

【样例输入】

3

10001011

【样例输出】

IBFBBBFIBFIIFF

【数据规模】

对于 40% 的数据， $N \leq 2$ ；

对于全部的数据， $N \leq 10$ 。

火星人

(martian.pas/dpr/c/cpp)

【问题描述】

人类终于登上了火星的土地并且见到了神秘的火星人。人类和火星人都无法理解对方的语言，但是我们的科学家发明了一种用数字交流的方法。这种交流方法是这样的，首先，火星人把一个非常大的数字告诉人类科学家，科学家破解这个数字的含义后，再把一个很小的数字加到这个大数上面，把结果告诉火星人，作为人类的回答。

火星人用一种非常简单的方式来表示数字——掰手指。火星人只有一只手，但这只手上有成千上万的手指，这些手指排成一列，分别编号为 1，2，3……。火星人的任意两根手指都能随意交换位置，他们就是通过这方法计数的。

一个火星人用一个人类的手演示了如何用手指计数。如果把五根手指——拇指、食指、中指、无名指和小指分别编号为 1，2，3，4 和 5，当它们按正常顺序排列时，形成了 5 位数 12345，当你交换无名指和小指的位置时，会形成 5 位数 12354，当你把五个手指的顺序完全颠倒时，会形成 54321，在所有能够形成的 120 个 5 位数中，12345 最小，它表示 1；12354 第二小，它表示 2；54321 最大，它表示 120。下表展示了只有 3 根手指时能够形成的 6 个 3 位数和它们代表的数字：

三进制数

123
132
213
231
312
321

代表的数字

1
2
3
4
5
6

现在你有幸成为了第一个和火星人交流的地球人。一个火星人会让你看他的手指，科学家会告诉你要加上去的很小的数。你的任务是，把火星人用手指表示的数与科学家告诉你的数相加，并根据相加的结果改变火星人手指的排列顺序。输入数据保证这个结果不会超出火星人手指能表示的范围。

【输入文件】

输入文件 `martian.in` 包括三行，第一行有一个正整数 N ，表示火星人手指的数目 ($1 \leq N \leq 10000$)。第二行是一个正整数 M ，表示要加上去的小整数 ($1 \leq M \leq 100$)。下一行是 1 到 N 这 N 个整数的一个排列，用空格隔开，表示火星人手指的排列顺序。

【输出文件】

输出文件 `martian.out` 只有一行，这一行含有 N 个整数，表示改变后的火星人手指的排列顺序。每两个相邻的数中间用一个空格分开，不能有多余的空格。

【样例输入】

5

3

1 2 3 4 5

【样例输出】

1 2 4 5 3

【数据规模】

对于 30% 的数据， $N \leq 15$ ；

对于 60% 的数据， $N \leq 50$ ；

对于全部的数据， $N \leq 10000$ ；

[1] 二叉树：二叉树是结点的有限集合，这个集合或为空集，或由一个根结点和两棵不相交的二叉树组成。这两棵不相交的二叉树分别称为这个根结点的左子树和右子树。

[2] 后序遍历：后序遍历是深度优先遍历二叉树的一种方法，它的递归定义是：先后序遍历左子树，再后序遍历右子树，最后访问根。

NOIP 普及组复赛参考程序
NOIP2004 普及组解题参考

第一题：不高兴的津津
方法：枚举
程序：

```

program unhappy; {written by lxq 2004.11.20}
var a,i,x,y,d,max : byte;
begin
  assign(input,'unhappy.in'); reset(input);
  assign(output,'unhappy.out'); rewrite(output);
  d := 0; max :=8;
  for i := 1 to 7 do begin
    readln(x,y);
    a := x+y;
    if a>max then
      begin
        max :=a; d := i;
      end;
    end;
    writeln(d);
    close(input); close(output);
  end.

```

第二题：花生采摘

方法：排个序，然后迭代递推

程序：

```

program peanuts; {written by lxq 2004.11.20}
type mytype=record
  x,y,d:integer;
end;
var time,all,num,i,j,m,n,k,u,v,z:integer;
  q:array[1..400] of mytype;
  t:mytype;
begin
  all:=0;
  assign(input,'peanuts.in');
  reset(input);
  readln(m,n,k);
  for i:=1 to m do
    begin
      for j:=1 to n do
        begin
          read(u);
          if u>0 then
            begin
              inc(all);
              q[all].x:=i;q[all].y:=j;q[all].d:=u;
              if all>1 then
                begin

```

```

        v:=1;
        while q[v].d>u do inc(v);
        t:=q[all];
        for z:=all downto v+1 do q[z]:=q[z-1];
        q[v]:=t;
    end;
end;
end;
readln;
end;
close(input);
num:=0;time:=0;u:=0;v:=q[1].y;
for i:=1 to all do
begin
    if time+abs(q[ i ].x-u)+abs(q[ i ].y-v)+1+q[ i ].x<=k
    then begin
        inc(num,q[ i ].d);
        time:=time+abs(q[ i ].x-u)+abs(q[ i ].y-v)+1;
        u:=q[ i ].x;v:=q[ i ].y;
    end
    else break;
end;
end;
assign(output,'peanuts.out');
rewrite(output);
writeln(num);
close(output);
end.

```

第三题 FBI 树

方法：递归即可，按后序遍历直接边生成边打印。

程序：

```

program fbi; {written by lxq 2004.11.20}
var f:array[1..1024] of char;
    i,k,n:integer;
    c:char;
function lastorder(i,j,n:integer):char;
var lc,rc:char;
begin
    if n=0 then lastorder:=f[ i ]
    else begin
        lc:=lastorder(i,(i+j) div 2,n-1);
        write(lc);
        rc:=lastorder((i+j) div 2+1,j,n-1);
        write(rc);
    end
end

```



```

                if lc=rc then lastorder:=lc else lastorder:='F';
            end;
        end;
    begin
        assign(input,'fbi.in');
        reset(input);
        readln(n);
        k:=1;
        for i:=1 to n do k:=k*2;
        for i:=1 to k do
            begin
                read(c);
                if c='0' then f[i]:='B' else f[i]:='I'
            end;
            readln;
        close(input);
        assign(output,'fbi.out');
        rewrite(output);
        writeln(lastorder(1,k,n));
        close(output);
    end.

```

第四题 火星人

方法：排列生成法，直接从指定序列用排列产生方法顺序生成到后面 M 个。

程序：

```

program martian; {written by lxq 2004.11.20}
const maxn=10000;
var a:array[1..maxn+1] of integer;
    b:array[1..maxn+1] of boolean;
    n,m,i,p,k:integer;
begin
    assign(input,'martian.in');
    reset(input);
    readln(n);
    readln(m);
    fillchar(b,sizeof(b),false);
    for i:=1 to n do begin read(a[i]);b[a[i]>:=true end;p:=n+1;
    k:=-1;
    while true do
        begin
            if p>n then begin
                dec(p);
                inc(k);
                b[a[k]>:=false;

```

```

        if k=m then break;
    end;
repeat inc(a[ i ] ); until not b[a[ i ] ]; b[a[ i ] ]:=true;
if a[ i ] >n then
    begin b[a[ i ] ]:=false;dec(p);b[a[ i ] >:=false end
else
    begin inc(p); a[ i ] :=0 end;
end;
assign(output,'martian.out');
rewrite(output);
for i:=1 to n-1 do write(a[ i ],' ');writeln(a[n]);
close(output)
end.

```