逻辑航线信息学奥赛系列教程

P1259 黑白棋子的移动

题目描述

有 2n个棋子排成一行,开始为位置白子全部在左边,黑子全部在右边,如下图为 n=5 的情况:

0000000000

移动棋子的规则是:每次必须同时移动相邻的两个棋子,颜色不限,可以左移也可以右移到空位上去,但不能调换两个棋子的左右位置。每次移动必须跳过若干个棋子(不能平移),要求最后能移成黑白相间的一行棋子。如 n=5 时,成为:

 $0 \bullet 0 \bullet 0 \bullet 0 \bullet 0 \bullet$

任务: 编程打印出移动过程。

输入格式

一个整数 n。

输出格式

若干行,表示初始状态和每次移动的状态,用"o"表示白子,"*"表示黑子,"-"表示空行。

输入样例

7

输出样例

0000000******

000000--******

000000******

00000--*****0*0*

00000*****--0*0*

0000--****0*0*

0000****--0*0*0*

000--***0*0*0*0

000*0**--*0*0*0* 0--*0**00*0*0*0*

0*0*0*--0*0*0*0*

--₀*₀*₀*₀*₀*₀*₀*

解析

编码

```
#include<bits/stdc++.h>
const int N = 1001;
using namespace std;
int n, cnt, space; //space始终指向相邻两个空格的第一个
char a[N];
//打印函数
void print() {
   //打印全部的状态
   for (int i = 1; i \le 2 * n + 2; i++)
       printf("%c", a[i]);
   printf("\n");
}
//将以m作为起始索引的棋子移动到以space作为起始的空白处
void move(int m) {
   a[space] = a[m];
   a[space + 1] = a[m + 1];
   a[m] = a[m + 1] = '-';
   //记录新的空白
   space = m;
  print();
}
//递归移动函数
//m为棋子数的一半
void mv(int m) {
   if (m == 4) //n==4相当于递归边界,按照固定步骤进行移动
   {
       move (4);
       move (8);
       move(2);
       move(7);
       move(1);
   } else //n>4时, 先移动两步达到n-1的状态, 重复递归, 直到n==4
       move (m);
      move (2 * m - 1);
       mv(m-1);
}
int main() {
```

```
scanf("%d", &n);
   //初始化白子、黑子、空位
   for (int i = 1; i <= n; i++) {
       a[i] = 'o';
   }
   for (int i = n + 1; i \le 2 * n; i++) {
      a[i] = '*';
   }
   for (int i = 2 * n + 1; i \le 2 * n + 2; i++) {
      a[i] = '-';
   }
   cnt = 0;
   //记录当前起始的空白索引
   space = 2 * n + 1;
   //打印初始化状态
  print();
  mv(n); //递归移动棋子
  return 0;
}
```

逻辑航线培优教育,信息学奥赛培训专家。

扫码添加作者获取更多内容。

