**题目描述】**

设有N个选手进行循环比赛，其中N=2M，要求每名选手要与其他N-1名选手都赛一次，每名选手每天比赛一次，循环赛共进行N-1天，要求每天没有选手轮空。

**【输入】**

输入：M。

**【输出】**

输出：表格形式的比赛安排表。一行各数据间用一个空格隔开。

**【输入样例】**

3

**【输出样例】**

1 2 3 4 5 6 7 8

2 1 4 3 6 5 8 7

3 4 1 2 7 8 5 6

4 3 2 1 8 7 6 5

5 6 7 8 1 2 3 4

6 5 8 7 2 1 4 3

7 8 5 6 3 4 1 2

8 7 6 5 4 3 2 1

**解题报告1**

先初始化为a[0][0]

一本通书上原理

|  |  |
| --- | --- |
|  | 右上等左方阵加一半 |
| 等右上方方阵 | 等左上方阵 |

代码书上有

**解题报告2**

和差不多

|  |  |
| --- | --- |
|  | 右上等左方阵加一半 |
| 左下等左方阵加一半 | 等左上方阵 |

核心代码

//填写右下角

 int i,j;

 for(i=0;i<M;i++)

 for(j =0;j<M;j++)

 a[i+M][j+M]=a[i][j];

 //填写右上角左下角

 for(i=0;i<M;i++)

 for(j =0;j<M;j++)

 a[i+M][j]=a[i][j+M]=a[i][j]+M;

//完善代码

#include<iostream>

using namespace std;

int b[10005][10005];

int f(int a[10005][10005],int M)

{

 //填写右下角

 int i,j;

 for(i=0;i<M;i++)

 for(j =0;j<M;j++)

 a[i+M][j+M]=a[i][j];

 //填写右上角左下角

 for(i=0;i<M;i++)

 for(j =0;j<M;j++)

 a[i+M][j]=a[i][j+M]=a[i][j]+M;

}

int p(int a[10005][10005],int M)

{

 int i,j;

 for(i=0;i<M;i++)

 {

 for(j =0;j<M;j++)

 cout<<a[i][j]<<" ";

 cout<<endl;

 }

}

int main()

{

 int i,j,N,a;

 cin>>a;

 N=1<<a;

 // cout<<;

 //int c[N];

 b[0][0]=1;

 for(i=1;i<=N/2;i\*=2)

 f(b,i);

 p(b,N);

 return 0;

}

**解题报告3**

规律也容易看出来

初始化

0 行 1 2 3 。。。N

1 奇偶交换1 2 1 4 3… N N-1

2 交换连续两个位置

4、奇偶交换

5、交换连续四个位置

6、奇偶交换

7、交换连续两个位置

8、奇偶交换

9、交换连续8个位置

10 奇偶交换

11 交换连续两个位置

12奇偶交换

13 交换连续4个位置

14、奇偶交换

15、交换连续2个位置

16、奇偶交换

17、交换连续16个位置

。

特别**提醒交换连接的N个位置是以前N行的顺序为参考**。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 规律 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |  |
| 0 | 0 | 1 | **2** | **3** | **4** | **5** | **6** | **7** | **8** | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |  |
| 1 | 0 | **2** | **1** | **4** | **3** | **6** | **5** | **8** | **7** | 10 | 9 | 12 | 11 | 14 | 13 | 16 | 15 |  |
| 2 | 2 | **3** | **4** | **1** | **2** | **7** | **8** | **5** | **6** | 11 | 12 | 9 | 10 | 15 | 16 | 13 | 14 |  |
| 3 | 0 | **4** | **3** | **2** | **1** | **8** | **7** | **6** | **5** | 12 | 11 | 10 | 9 | 16 | 15 | 14 | 13 |  |
| 4 | 4 | **5** | **6** | **7** | **8** | **1** | **2** | **3** | **4** | 13 | 14 | 15 | 16 | 9 | 10 | 11 | 12 |  |
| 5 | 0 | **6** | **5** | **8** | **7** | **2** | **1** | **4** | **3** | 14 | 13 | 16 | 15 | 10 | 9 | 12 | 11 |  |
| 6 | 2 | **7** | **8** | **5** | **6** | **3** | **4** | **1** | **2** | 15 | 16 | 13 | 14 | 11 | 12 | 9 | 10 |  |
| 7 | 0 | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 |  |
| 8 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |  |
| 9 | 0 | 10 | 9 | 12 | 11 | 14 | 13 | 16 | 15 | 2 | 1 | 4 | 3 | 6 | 5 | 8 | 7 |  |
| 10 | 2 | 11 | 12 | 9 | 10 | 15 | 16 | 13 | 14 | 3 | 4 | 1 | 2 | 7 | 8 | 5 | 6 |  |
| 11 | 0 | 12 | 11 | 10 | 9 | 16 | 15 | 14 | 13 | 4 | 3 | 2 | 1 | 8 | 7 | 6 | 5 |  |
| 12 | 4 | 13 | 14 | 15 | 16 | 9 | 10 | 11 | 12 | 5 | 6 | 7 | 8 | 1 | 2 | 3 | 4 |  |
| 13 | 0 | 14 | 13 | 16 | 15 | 10 | 9 | 12 | 11 | 6 | 5 | 8 | 7 | 2 | 1 | 4 | 3 |  |
| 14 | 2 | 15 | 16 | 13 | 14 | 11 | 12 | 9 | 10 | 7 | 8 | 5 | 6 | 3 | 4 | 1 | 2 |  |
| 15 | 0 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |  |
| 16 | 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**AC代码如下：**

#include<iostream>

#include<cmath>

using namespace std;

int b[10005][10005];

//奇偶交换

int px(int a[10005][10005],int M,int row)

{

 int i,j,t;

 for(i=1;i<M;i+=2)

 {

 a[row][i-1]=a[row-1][i];

 a[row][i]=a[row-1][i-1];

 }

}

//输出二维数据

int p(int a[10005][10005],int M)

{

 int i,j;

 for(i=0;i<M;i++)

 {

 for(j =0;j<M;j++)

 cout<<a[i][j]<<" ";

 cout<<endl;

 }

}

//初始化1－N二维数组第0行代码

int ps(int a[10005][10005],int M,int r)

{

 int i,j;

 for(i=0;i<M;i++)

 a[r][i]=i+1;

}

//核心代码

//连续交换N个位置是以前N行为参考依据，old为前N行，

//now为当前要填充连续交换N位置，这时用m来代表连续交换m个数

int fx(int a[10005][10005],int N,int m,int old,int now)

{

 int i,j,t;

 for(i =0;i<N;i+=2\*m)//以m为中心左右交换

 {

 j=i;

 while(j<m+i)

 {

 a[now][j+m]=a[old][j];

 a[now][j]=a[old][j+m];

 j++;

 }

 }

}

//能被2整除的次数

//连续交换次数的判断，规律不好总结，也是种解题的难度之一

int ftwo(int a)

{

 int t =0;

 while(a%2==0)

 {

 a/=2;

 t++;

 }

 if(t==0)

 return 0;

 else return pow(2,t);

}

int main()

{

 int i,j,N,a;

 cin>>a;

 N=1<<a;

 // cout<<;

 //int c[N];

 ps(b,N,0);

 // for(i=1;i<=N/2;i\*=2)

 // f(b,i);

 for(i = 1;i<N;i++)

 {

 int t = ftwo(i);//交换数量

 if(t==0)

 px(b,N,i);//<<endl;

 else

 fx(b,N,t,i-t,i);

 }

 p(b,N);

 return 0;

}