【实验要求】

设有 n 位运动员要进行网球循环赛。设计一个满足下列条件的比赛日程表：

要求：每个选手必须与其他 n-1 个选手各赛一次；

每个选手一天只能赛一次；

当 n 是偶数时，循环赛进行 n-1 天。

当 n 是奇数时，循环赛进行 n 天。

【主要思想】

采用分治法，

首先，我们来解释示例的矩阵

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

第一列代表选手，后面的第 n 列代表的是在第 n-1 天遭遇的对手。

要解决这个问题，我们使用分治法。N 位选手的比赛日程表就可以通过为 n/2 个选手设计的比赛日程表来决定。

1、当参赛选手为两个人的时候，矩阵很好生成，即为

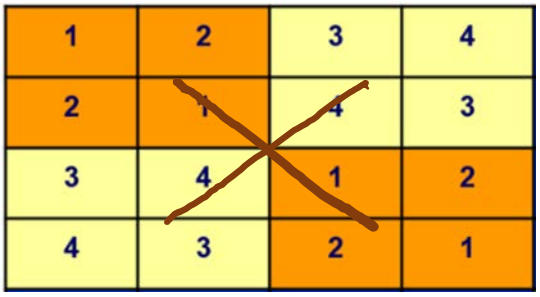
1	2
2	1

2、当参赛选手为 2^2 时，第二天、第三天的时候，1 号 2 号选手已经比完了，可以与 3 号比赛，所以此时的表格如下

1	2		
2	1		
		1	2
		2	1

3、同理我们可以得到

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1



不难看出，这个表格是通过交叉填充的方式得到的，也就是说四个选手的结果，由两个选手的结果来决定。由此，形成了分治的算法。由 4 到 8 的过程迎刃而解

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

4、一旦遭遇奇数位选手怎么做？

我们可以给多出来的这个选手一个虚拟的对手，再按照偶数个选手进行处理。此时要对矩阵做如下处理。以三位选手为例：

（1）假设有四位选手，得到如下矩阵，但是 4 号选手并不存在，于是删除最后一行。

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

1	2	3	4
2	1	4	3
3	4	1	2

（2）由于 4 号选手不存在，于是将 4 置为 0，表示轮空

1	2	3	0
2	1	0	3
3	0	1	2

这样我们也能看出，选手数 n 为奇数时，需要 n 天，符合要求。

整体上看，该问题的解决采用了分治法的思想，我的实现涉及到了递归的调用。

【编程实现】

环境：VS2017、g++

语言：C++

```
#include<iostream>
#include<vector>
#include<cstdlib>

using namespace std;

void show(vector<vector<int> > a,int size)
{
    int i, j;
    cout << "Number before\":\"means the player. After the \":\" means the
player's opponent in No.i day.\n";
    cout << "\t";
    for (j = 0; j <= a[0].size() - 2; j++)
        cout << "Day" << j + 1 << "\t";
    cout << endl;
    for (i = 0; i <= size - 1; i++)
    {
        for (j = 0; j <= a[i].size() - 1; j++)
        {
            if (j == 0)
                cout << a[i][j] << ":\t";
            else
                cout << a[i][j] << "\t";
        }
        cout << endl;
    }
    cout << endl;
}

void replaceVirtual(vector<vector<int> > &a, int m)
{
    int i, j;
    for (i = 0; i<a.size() - 1; i++)
    {
        for (j = 0; j <= a[0].size() - 1; j++)
        {
            if (a[i][j] == m)
                a[i][j] = 0;
```

```
        }
    }
}

void copyeven(vector<vector<int> > &a, int m)
{
    int i, j;
    for (j = 0; j<m; j++)
    {//求第2组的安排
        for (i = 0; i<m; i++)
            a[i + m][j] = a[i][j] + m;
    }
    for (j = m; j<2 * m; j++)//两组间比赛的安排
    {
        for (i = 0; i<m; i++)
            a[i][j] = a[i + m][j - m];//把左下角拷贝到右上角
        for (i = m; i<2 * m; i++)
            a[i][j] = a[i - m][j - m];//把左上角拷贝到右下角
    }
}

void copyodd(vector<vector<int> > &a, int m)
{
    int i, j;
    for (j = 0; j <= m; j++)
    {//求第2组的安排
        for (i = 0; i<m; i++)
        {
            if (a[i][j] != 0)
                a[i + m][j] = a[i][j] + m;
            else
            {//两个队各有一名选手有空
                a[i + m][j] = i + 1;
                a[i][j] = i + m + 1;
            }
        }
    }

    for (i = 0, j = m + 1; j<2 * m; j++)
    {//安排两组选手
        a[i][j] = j + 1;//1号选手的后m-1天的安排
        a[(a[i][j] - 1)][j] = i + 1;//对手后m-1天的安排
    }
```

```
    //先安排1号的赛程
    for (i = 1; i<m; i++)
    {//其他选手的后m-1天的安排
        for (j = m + 1; j<2 * m; j++)
        {//向下m+1~2*m循环递增
            a[i][j] = ((a[i - 1][j] + 1) % m == 0) ? a[i - 1][j] + 1 : m +
(a[i - 1][j] + 1) % m;//对应组的对手也要做相应的安排
            a[(a[i][j] - 1)][j] = i + 1;
        }
    }
}

void copy(vector<vector<int> > &a, int m)
{
    if (m % 2 == 1)
        copyodd(a, m / 2);
    else
        copyeven(a, m / 2);
}

void tournament(vector<vector<int> > &a, int num)
{
    if (num == 1)
    {
    }
    else
    {
        if (num % 2 == 0)
        {//偶数
            tournament(a, num / 2);
            copy(a, num);
        }
        else
        {//奇数
            tournament(a, num + 1);
            replaceVirtual(a, num + 1);
        }
    }
}

void schedule(vector<vector<int> > &a, int num)
{
    int i;
    for (i = 0; i <= a.size() - 1; i++)
```

```
        a[i][0] = i + 1;
    tournament(a, num);
}

int main()
{
    int i, j,size;
    vector<vector<int> > a;
    cout << "Please input the number of players:_\b";
    cin >> size;
    while (size != 0)
    {
        for (i = 0; i <= a.size() - 1; i++)
        {
            if (size % 2 == 0)
            {
                a.resize(size);
                a[i].resize(size);
            }
            else if (size % 2 == 1)
            {
                a.resize(size+1);
                a[i].resize(size + 1);
            }
        }
        for (i = 0; i <= a.size() - 1; i++)
            for (j = 0; j <= a[i].size() - 1; j++)
                a[i][j] = 0;

        schedule(a, size);
        show(a,size);
        cout << "Please input the number of players(Press 0 to exit):_\b";
        cin >> size;
    }
    system("pause");
    return 0;
}
```

【使用方法】

输入人数即得结果，可以重复使用，直到输入为0。

【实验结果截图】（我的控制台有一点点的透明。。）

【遇到的问题】

（1）利用vector来定义二维数组的时候，通过查阅资料得知模板类可以嵌套定义。

（2）我在处理奇数位选手时模拟了偶数位选手的情况，但是最初申请空间的时候却没有分配偶数位的空间，导致一直出bug，找了好久才发现问题，多亏VS2017可以提示可能出现问题的位置。

（3）处理奇数的时候，要考虑到更多的因素，花费了很多时间。