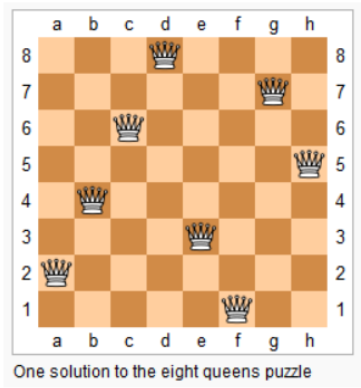


DFS (2)

n -皇后问题是指将 n 个皇后放在 $n \times n$ 的国际象棋棋盘上，使得皇后不能相互攻击到，即任意两个皇后都不能处于同一行、同一列或同一斜线上。



现在给定整数 n ，请你输出所有的满足条件的棋子摆法。

输入格式

共一行，包含整数 n 。

输入格式

共一行，包含整数 n 。

输出格式

每个解决方案占 n 行，每行输出一个长度为 n 的字符串，用来表示完整的棋盘状态。

其中 `.` 表示某一个位置的方格状态为空，`Q` 表示某一个位置的方格上摆着皇后。

每个方案输出完成后，输出一个空行。

注意：行末不能有多余空格。

输出方案的顺序任意，只要不重复且没有遗漏即可。

数据范围

$1 \leq n \leq 9$

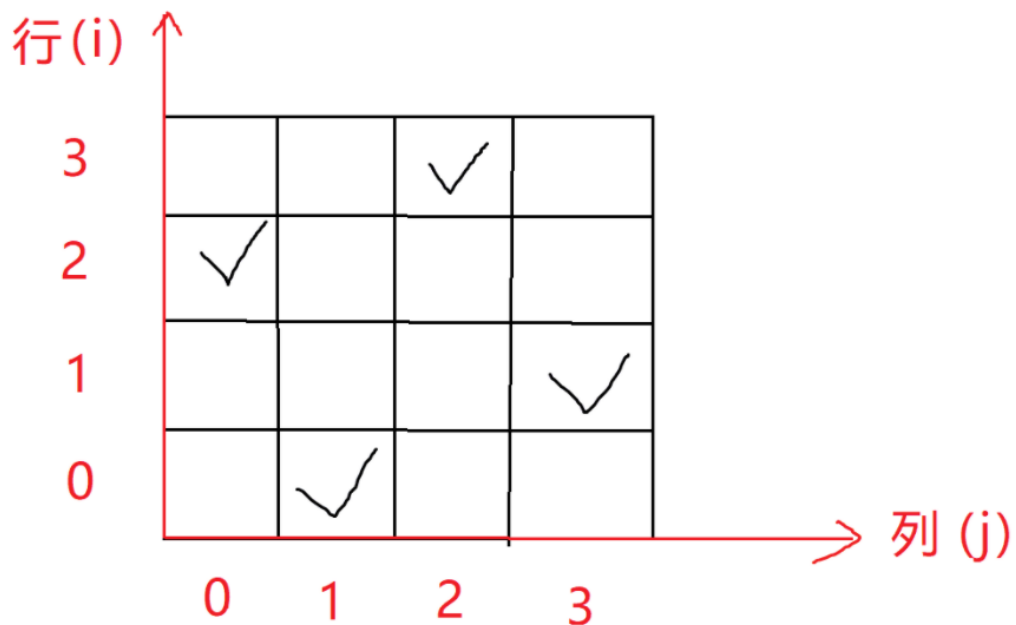
输入样例：

4

输出样例：

```
.Q..
...Q
Q...
..Q.

..Q.
Q...
...Q
.Q..
```



计算出每个遍历点的两条斜对线的函数

设该点坐标为 (j, i)

左斜线函数 $i = -j + c1$ ($c1$ 为参数) 即 $i + j = c1$

右斜线函数 $i = j + c2$ ($c2$ 为参数) 即 $i - j = c2$

(为了方便存储, 将 $i - j + n$ 记为 $c2$ 的映射值)

y总代码:

```
//参考DFS (1)
#include <iostream>

using namespace std;

const int N = 10;

int n;
bool row[N], col[N], dg[N * 2], udg[N * 2];
char g[N][N];

void dfs(int x, int y, int s)
{
    if (s > n) return;
    if (y == n) y = 0, x ++ ;

    if (x == n)
    {
        if (s == n)
        {
            for (int i = 0; i < n; i ++ ) puts(g[i]);
            puts("");
        }
        return;
    }

    g[x][y] = '.';
    dfs(x, y + 1, s);
```

```

    if (!row[x] && !col[y] && !dg[x + y] && !udg[x - y + n])
    {
        row[x] = col[y] = dg[x + y] = udg[x - y + n] = true;
        g[x][y] = 'Q';
        dfs(x, y + 1, s + 1);
        g[x][y] = '.';
        row[x] = col[y] = dg[x + y] = udg[x - y + n] = false;
    }
}

int main()
{
    cin >> n;

    dfs(0, 0, 0);

    return 0;
}

```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/47097/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

//逐个遍历，时间复杂度高

```
#include <iostream>
```

```
using namespace std;
```

```
const int N = 20;
```

```
int n;
```

```
char g[N][N];
```

```
bool col[N], dg[N], udg[N];
```

```
void dfs(int u)
```

```
{
```

```
    if (u == n)
```

```
    {
```

```
        for (int i = 0; i < n; i ++ ) puts(g[i]);
```

```
        puts("");
```

```
        return;
```

```
    }
```

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

```
        if (!col[i] && !dg[u + i] && !udg[n - u + i])
```

```
        {
```

```
            g[u][i] = 'Q';
```

```
            col[i] = dg[u + i] = udg[n - u + i] = true;
```

```
            dfs(u + 1);
```

```
            col[i] = dg[u + i] = udg[n - u + i] = false;
```

```
            g[u][i] = '.';
```

```
        }
```

```
}
```

```
int main()
```

```
{
```

```
cin >> n;
for (int i = 0; i < n; i ++ )
    for (int j = 0; j < n; j ++ )
        g[i][j] = '.';

dfs(0);

return 0;
}
```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/47097/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。