

计算机算法设计与实践-课后作业5

181002222 连月菡

计算机算法设计与实践-课后作业5

[题目描述](#)

[解答](#)

题目描述

修改并实现算法，要求输出n 皇后问题的所有解。

解答

原始算法:

```
1  #include <iostream>
2
3  using namespace std;
4  int x[1000001];
5
6  int Place(int k)
7  {
8      for (int i = 0; i < k; ++i)
9      {
10         if (x[i] == x[k] || abs(i - k) == abs(x[i] - x[k]))
11             return 1;
12     }
13     return 0;
14 }
15
16 void Queen(int n)
17 {
18     int k = 0;
19     while (k >= 0)
20     {
21         x[k]++;
22         while (x[k] < n && Place(k) == 1)
23             x[k]++;
24         if (x[k] < n && k == n - 1)
25         {
26             for (int i = 0; i < n; ++i)
27                 cout << x[i] + 1 << " ";
28             cout << endl;
29             return;
30         }
31         else if (x[k] < n && k < n - 1)
32             k += 1;
33         else x[k--] = -1;
```

```

34     }
35     cout << "无解" << endl;
36 }
37
38 int main()
39 {
40     int n;
41     Queen(n);
42 }

```

```

3
无解

```

输入 3,得到无解

```

Microsoft Visual Studio 调试控制台
4
3 1 4 2

```

输入 4 得到一个结果

```

Microsoft Visual Studio 调试控制台
8
2 4 6 8 3 1 7 5

```

输入8 得到一个结果

修改后:

```

1  #include <iostream>
2  using namespace std;
3  #define num 3      //数目
4  #define INIT -1000 //初始值
5  int a[num];        //棋盘
6  void print()        //打印解
7  {
8      int i;
9      for (i = 0; i < num; ++i)
10         printf("%d ", a[i]);
11     printf("\n");
12 }
13
14
15 int Place(int row, int col)    //判断第row行第col列是否可以放置皇后
16 {
17     for (int i = 0; i < num; ++i)    //对棋盘进行扫描
18     {
19         if (a[i] == col || abs(i - row) == abs(a[i] - col))    //判断列和斜线是
20             否有冲突
21             return 0;
22     }
23     return 1;
24 }
25 void Queen()
26 {

```

```

27     bool flag = 0; //判断到底有没有
28     int n = 0;
29     int i = 0, j = 0;
30     while (i < num)
31     {
32         while (j < num) //对i行的每一列进行查看, 看是否可以放置皇后
33         {
34             if (Place(i, j)) //可以放置皇后
35             {
36                 a[i] = j; //第i行放置皇后
37                 j = 0; //第i行放置皇后确定下一行的皇后位置, 所以此处将j清
零, 从下一行的第0列开始逐列查看
38                 break;
39             }
40             else ++j; //返回值为0, 说明不行, 继续查看下一列
41         }
42         if (a[i] == INIT) //第i行没有解
43         {
44             if (i == 0) //回溯到第一行, 说明已经找到所有的解
45                 break;
46             else //没有找到可以放置皇后的列, 回溯
47             {
48                 --i;
49                 j = a[i] + 1; //把上一行皇后的位置往后移一列
50                 a[i] = INIT; //把上一行皇后的位置清除, 重新探测
51                 continue;
52             }
53         }
54         if (i == num - 1)
55         {
56             flag = 1;
57             printf("%d : \n", ++n); print(); //最后一行找到了位置, 打印结果
58             j = a[i] + 1; //从最后一行放置皇后列数的下一列继续探测
59             a[i] = INIT; //清除最后一行的皇后位置
60             continue;
61         }
62         ++i; //继续查看下一行的皇后位置
63     }
64     if (!flag)
65         cout << "无解" << endl;
66 }
67
68 int main(void)
69 {
70     for (int i = 0; i < num; ++i) //初始化
71         a[i] = INIT;
72     Queen();
73     return 0;
74 }

```

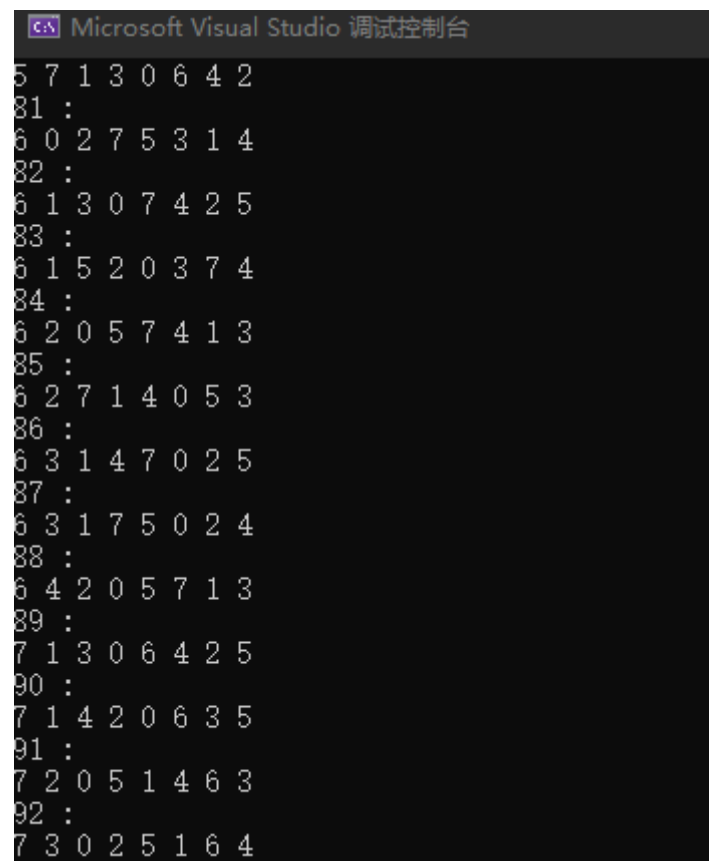
修改后, 若 num=4, 则输出以下2个结果:

```

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

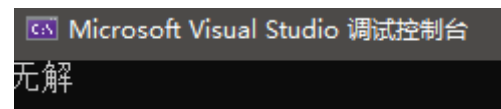
```

若 num=8, 则输出以下92个结果:



```
Microsoft Visual Studio 调试控制台
5 7 1 3 0 6 4 2
81 :
6 0 2 7 5 3 1 4
82 :
6 1 3 0 7 4 2 5
83 :
6 1 5 2 0 3 7 4
84 :
6 2 0 5 7 4 1 3
85 :
6 2 7 1 4 0 5 3
86 :
6 3 1 4 7 0 2 5
87 :
6 3 1 7 5 0 2 4
88 :
6 4 2 0 5 7 1 3
89 :
7 1 3 0 6 4 2 5
90 :
7 1 4 2 0 6 3 5
91 :
7 2 0 5 1 4 6 3
92 :
7 3 0 2 5 1 6 4
```

若num=3 则无解



```
Microsoft Visual Studio 调试控制台
无解
```