数据结构课程设计

题目四 N皇后问题

文档

班级：软件5班

学号：1353010

姓名：薛梦迪

**一、项目简介：**

八皇后问题是一个古老而著名的问题，是回溯算法的经典问题。该问题是十九世纪著名的数学家高斯在1850年提出的：在8\*8的国际象棋棋盘上，安放8个皇后，要求没有一个皇后能够“吃掉”任何其它一个皇后，即任意两个皇后不能处于同一行，同一列或者同一条对角线上，求解有多少种摆法。

高斯认为有76种方案。1854年在柏林的象棋杂志上不同的作者发表了40种不同的解，后来有人用图论的方法得到结论，有92中摆法。

本实验拓展了N皇后问题，即皇后个数由用户输入。

**二、项目功能：**

1、皇后个数的输入。

2、用回溯算法来解决问题。

**三、项目代码说明：**

#include"iostream"

#include<math.h>

using namespace std;

判断位置是否合理的函数。即确保下一行的皇后不在前面行的同一列或对角线上。

bool isPlaceOk(int k,int \*x){

int j;

for (j = 0; j < k; j ++){

if (x[j] == x[k] || abs(j - k) == abs(x[j] - x[k])){

return false;

}

}

return true;

}

打印皇后分布的函数。x[i]记录第i行的皇后所在位置，所以执行两个循环来打印出N×N皇后矩阵，并且记录有多少种解。

void print(int num,int \*x,int &sum){

int i,j;

for(i=0;i<num;i++){

for(j=0;j<num;j++){

if(j==x[i]){

cout << "X ";

}else{

cout << "0 ";

}

}

cout << endl;

}

cout << endl;

sum++;

}

回溯算法函数。K代表列数，先循环判断位置是否合理，若不合理则此列皇后向后移一格，若合理则进入if判断，if判断中，如果此行可以放置皇后，且进行到了最后一行，则打印整个矩阵，否则进行下一行，若此行不能放置皇后，则回溯到上一行，并将皇后向右移一格。

void track(int num,int \*x,int &sum){

int k = 0;

x[0] = -1;

while(k>=0){

x[k]++;

while((x[k] < num) && !(isPlaceOk(k,x))){

x[k]++;

}

if(x[k] < num){

if(k == num - 1){

print(num,x,sum);

}else{

x[++k]=-1;

}

}else{

k--;

}

}

}

int main(){

cout << "现有N×N的棋盘，放入N个皇后，要求所有皇后不在同一行、列和同一斜线上！" << endl << endl;

cout << "请输入皇后的个数：";

主函数。根据用户输入的皇后个数来创建数组并执行回溯算法函数。

int number;

cin >> number;

int x[number];

int sum = 0;

track(number,x,sum);

cout << number << "皇后问题有" << sum << "个解" << endl;

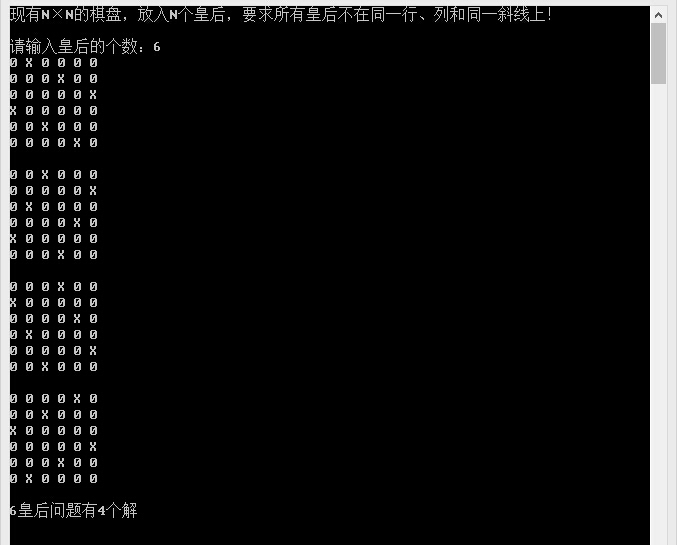
getchar();

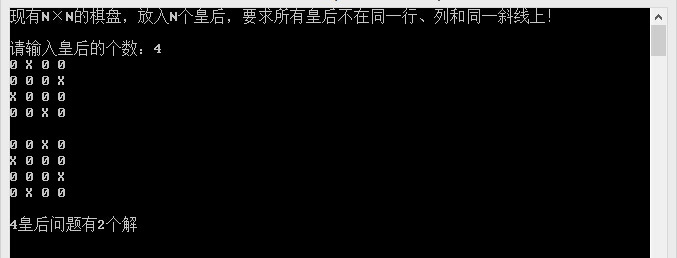
getchar();

return 0;

}

**四、项目运行的界面：**





输入皇后个数，则自动列出解和解的个数。