```
// CS211
#include <iostream>
#include<cstdlib>
#include <cmath>
using namespace std;


bool ok(int q[], int col){
   for(int i=0; i<col; i++)
      if(q[col]==q[i] || (col-i)==abs(q[col]-q[i])) return false;

    return true;
};

void backtrack(int &col){
    col--;
       if(col==-1)
              {system("PAUSE"); exit(1);}
};

void print(int q[]){
    static int count =0;
    cout<< ++ count<<endl<<endl;
    int i;
    for(i=0; i<8; i++)
        cout<<q[i]<<" ";
     cout<<endl<<endl;
};


 void main(){
    int q[8]; q[0]=0;
       int c=1;

    // the boolean variable "from_backtrack" keeps track if we need
    // to reset the row to the top of the current column or not.

    bool from_backtrack=false;


    // The outer loop keeps looking for solutions
    // The program terminates from function backtrack
    // when we are forced to backtack into column -1
    while(true){
        while(c<8){

            // if we just returned from backtrack,use current value of q[c]
            // if not, get ready to start at the top of this column
            if(!from_backtrack)
               code goes here     //start at the top


            from_backtrack=______; // reset for the next time through

            while(q[c]<8){ // place queen in this column or backtrack as required
                code goes here

                // if the row = 8, there is no valid square in this column
                // so backtrack and continue the loop in the previous column
                if(code goes here) {code goes here}

                //if this position is ok, place the queen
                // and move on to the next column,
                // otherwise keep looking in this column
                code goes here
            }
            code goes here// placed ok, move to the next column
        }
        // one complete solution found, print it.
        code goes here // board completed, print it out
        code goes here //find the next place for the queen, going back as far as need be
        code goes here
    }
}
```