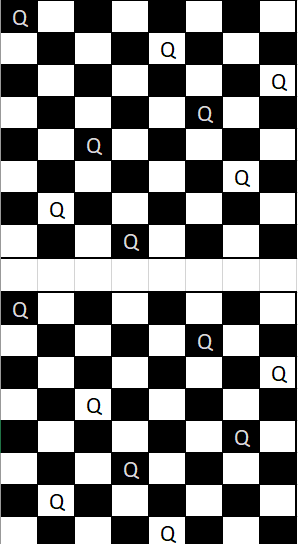


|  |  |
| --- | --- |
| **Ch13-02: *8 Queens*** | |
| **Line#** | **Code** |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | #include <iostream>  using namespace std;  int Qs[8];  int Row = 0;  int NoOfAnswers = 0;  bool IsValidColumn(int c) {  for (int r = 0; r < Row; r++) {  //Vertical Check  if (Qs[r] == c) return false;  //Diagonal check  int dR = Row - r; //Alway +ve  int dC = c - Qs[r]; //+ve or -ve  if (dC < 0) dC = -dC;  if (dC == dR) return false;  }  return true;  }  void ShowAnswer() {  for (int r = 0; r < 8; r++) {  for (int c = 0; c < 8; c++) cout << ((Qs[r] == c) ? 'Q' : '.');  cout << endl;  }  cout << endl;  }  void Solve() {  if (Row < 8) {  for (int c = 0; c < 8; c++) {  if (IsValidColumn(c)) {  Qs[Row] = c;  Row++;  Solve(); //Recursion  Row--;  }  }  }  else {  NoOfAnswers++;  ShowAnswer();  }  }  int main() {  Solve();  cout << "No of Answers is " << NoOfAnswers << endl;  return 0;  } |

**Sample Answers:**



|  |  |
| --- | --- |
| Student.h | |
| Line# | Code |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11** | #pragma once  **class** Student{  **public**:  **const char**\* Name;  **unsigned char** Age;  **float** CGPA;  weak\_ptr<Student> LearnFrom;  Student(**const char**\* name, **unsigned char** age, **float** cgpa);  Student();  ~Student();  }; |