|  |  |
| --- | --- |
| **Ch13-01:** *Magic3x3* | |
| **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  51  52  53  54  55  56  57  58  59  60  61 | #include <iostream>  using namespace std;  int Slots[9];  int Level = 0;  int NoOfPatterns = 0;  int NoOfAnswers = 0;  void ShowAnswer() {  cout << Slots[0] << Slots[1] << Slots[2] << endl;  cout << Slots[3] << Slots[4] << Slots[5] << endl;  cout << Slots[6] << Slots[7] << Slots[8] << endl;  cout << endl;  }  bool IsNotInused(int n) {  for (int i = 0; i < Level; i++) if (Slots[i] == n) return false;  return true;  }  //012  //345  //678  bool IsAnswer() {  int H1 = Slots[0] + Slots[1] + Slots[2];  int H2 = Slots[3] + Slots[4] + Slots[5];  int H3 = Slots[6] + Slots[7] + Slots[8];  int V1 = Slots[0] + Slots[3] + Slots[6];  int V2 = Slots[1] + Slots[4] + Slots[7];  int V3 = Slots[2] + Slots[5] + Slots[8];  int D1 = Slots[0] + Slots[4] + Slots[8];  int D2 = Slots[2] + Slots[4] + Slots[6];  return (H1 == H2) && (H1 == H3) &&  (H1 == V1) && (H1 == V2) && (H1 == V3) && (H1 == D1) && (H1 == D2);  }  void Solve() {  if (Level < 9) {//Generate  for (int n = 1; n < 10; n++) {  if (IsNotInused(n)) {  Slots[Level] = n;  Level++;  Solve(); //Recursion  Level--; //Backtracking  }  }  }  else {//Test  NoOfPatterns++;  if (IsAnswer()) {  NoOfAnswers++;  ShowAnswer();  }  }  }  int main() {  Solve();  cout << "No of answers found is " << NoOfAnswers << endl;  cout << "No of Patterns examined is " << NoOfPatterns << endl;  return 0;  } |