|  |
| --- |
| **Ch13-01:** *Magic3x3* |
| **Line#** | **Code** |
| 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061 | #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//678bool 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;} |