Muhammad Yassa - 8 Integers in a cross pattern program with backtracking. Short Report/ Summary.

Wednesday, 11 October 2023 7:59 PM

The program makes use of backtracking method and check functions to solve the 8 integers in a cross problem. The problem requires us to allocate integers 1-8 to the squares in a cross pattern, with the restriction that no two adjacent squares contain consecutive integers, adjacent meaning vertically, horizontally, and diagonally. The code outputs all 4 solutions in a cross pattern. We initialize an array q to represent the grid boxes, and initialize a 2d helper array which contains all the grid boxes that have to be checked for each column. We place the integer 1 in the very first grid box and use two while loops to increment between columns and integers 1-8, the outer while loop checks If column number has reached 8, meaning all columns have been filled, and calls the print function to print out the solution and then we backtrack. If column number has not reached 8, we set the value of q[c] to 0. The inner while loop increments the integers from 1-8 , an if function is used to check if the value of the column has reached 9, meaning none of the integers from 1-8 has passed the check function, so we backtrack. The integer passes all checks, where we check if the integer is being repeated and If the integer is not consecutive to its adjacent squares, if the absolute value of the difference between the current column value and the column value we're checking is one, the check function returns false. If an integer passes all checks we break inner while loop and move to the next column. Program keeps backtracking till all possible solutions are found and printed, until column number goes below 0 and both while loops are exited and the program returns with code 0.

Solution #1:
35
7182
46
Solution #2:
46
7182
35
Solution #3:
53
2817
64
Solution #4:
64
2817
53