Student’s Name: Mubasshir Al Shahriar

Assignment: Redo the 8 queens 1 dimensional array program with backtracking WITHOUT gotos.

Short Report/Summary:

* Initialized 1D array q[8] and a integer variable c to initialize a chessboard, where c represents the column and q[c] represents the row. Started from first column, first row.
* Used 2 while loops (nested) in the program, where the outer one was used to go to the next column and to go to the printing part of the program by calling the print function which was created previously to be used in the main function to print the solved array. According to the condition, this outer while loop will be ended when it backtracks from the first column, which means, when all possible solutions are found.
* The inner while loop was used to go to the next row and to check if that row is suitable to place a queen by calling the ok function. If the ok function returns true, which means, if the position is suitable, then the inner while loop gets terminated and it goes back to the beginning of the outer while loop and then prints the result and goes for backtracking afterwards.
* **In simple words, starting from the first row, first column, the program goes to the next column of the chessboard through outer while loop. Then, if the program has already passed the last column, it prints the found solution and starts backtracking for another set of solution. If not, then it enters the inner while loop, starting from the first row of that column, it calls the previously created ok function, that runs 3 tests (row test, up diagonal test and down diagonal test) and checks if it is ok to place a queen in that row. If it says the position is ok, then the program goes back to the beginning of the outer loop and prints the result. Otherwise, inside the inner loop, it just goes to the next row and keeps checking the position with ok function until it finds a suitable position, or it reaches the last row of the column. In case it reaches the last row of the column, yet couldn’t find a place, it backtracks and goes to the previous column to change the previous position of queen in that column. The outer while loop gets terminated when it successfully finds all the (92) possible solutions and then the program ends.**

Comment: This is my algorithm to place 8 queens in a chessboard so none of them threat each other using 1D array without goto statement and using nested while loop. This is different from the previous assignment where we solved it without goto in 1D format. Because, at that previous one, we used 8 nested for loops to assign and check all the possible result of an 8x8 chessboard with 8 queens. That one was very inefficient as the program had to check all the solutions which was unnecessary. But in this program, using 2 while loops (nested) it goes column by column and checks row by row, and as soon as it finds a position, it doesn’t check the next rows unnecessarily, which makes this program more efficient than the previous one. I compiled and ran the program; it ran correctly and successfully printed 92 correct solutions in output.

Screenshots of Output:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated