

Project Report

N-Queens Visualizer in C++

By : Ashif Ahamad

**Table of Contents**

1. Introduction

2. Project Objectives

3. Literature Review

4. System Design

5. Implementation

6. Testing

7. Conclusion

8. Future Work

9. References

# Introduction

## Overview

The N-Queens problem is a well-known combinatorial problem in which the objective is to place N queens on an N×N chessboard such that no two queens can attack each other. This project aims to solve the N-Queens problem and visualize the solution in a console-based application using C++.

## Project Objectives

* Implement an efficient algorithm to solve the N-Queens problem.
* Develop a console-based application to visualize the placement of queens on the board.
* Enhance the visual presentation using colors and formatted console output.
* Ensure the application is user-friendly and interactive.

# Literature Review

## N-Queens Problem

The N-Queens problem has been extensively studied in computer science and mathematics. It is an example of a constraint satisfaction problem and can be solved using various approaches such as backtracking, heuristic search, and genetic algorithms.

## Visualization Techniques

Console-based applications can utilize various techniques to enhance visual presentation, including ASCII art, ANSI escape codes for color, and formatted output.

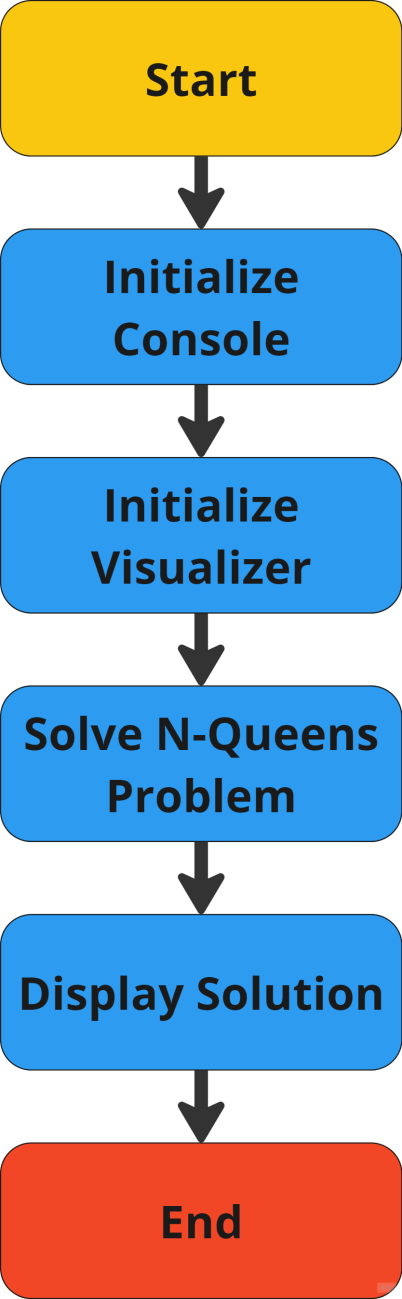
# System Design

## Architecture

The system consists of the following components:

* **Console Initialization**: Sets up the console environment for UTF-8 encoding and determines the console width.
* **Visualizer Initialization**: Displays a welcome message and prompts the user for input.
* **Solver**: Implements the backtracking algorithm to solve the N-Queens problem.
* **Visualizer**: Displays the board with the queens placed according to the solution

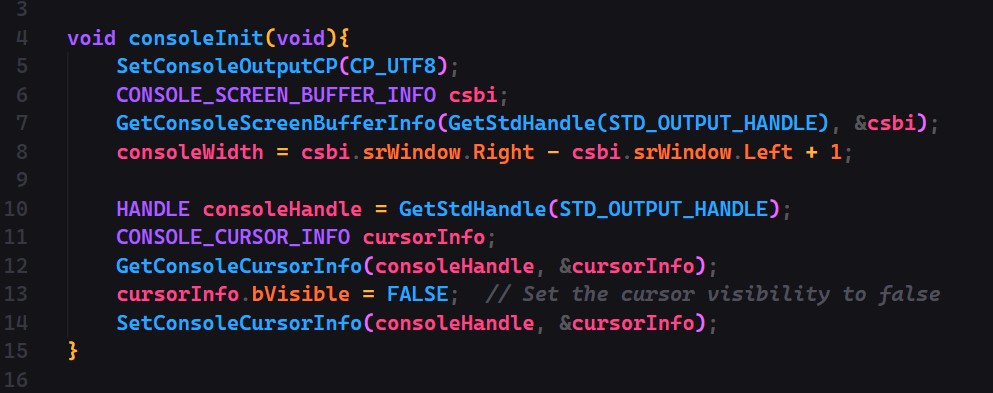
# Flowchart



# Implementation

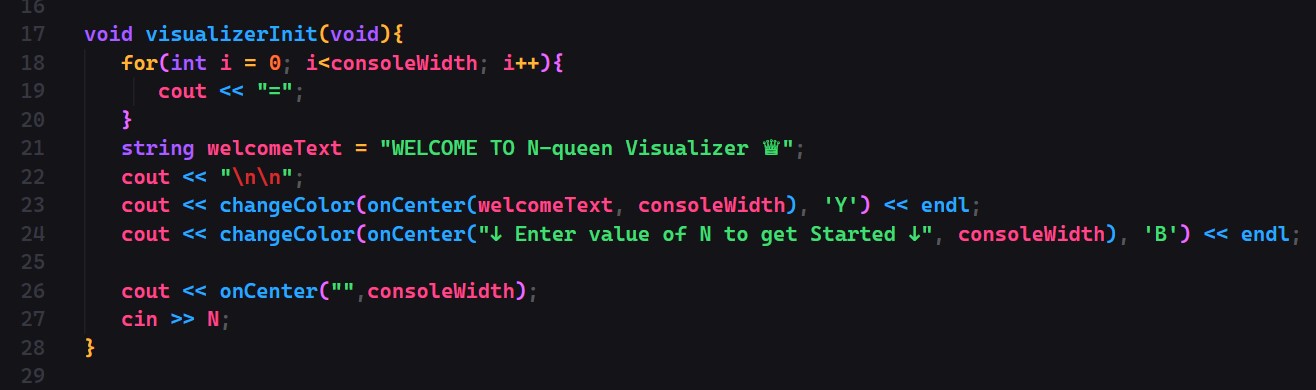
## Console Initialization

The *consoleInit* function sets the console to UTF-8 encoding, determines the width of the console for proper formatting and sets the cursor visibility to false.



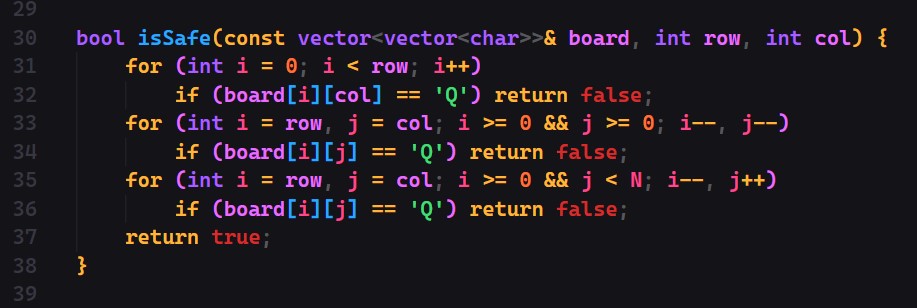
## Visualizer Initialization

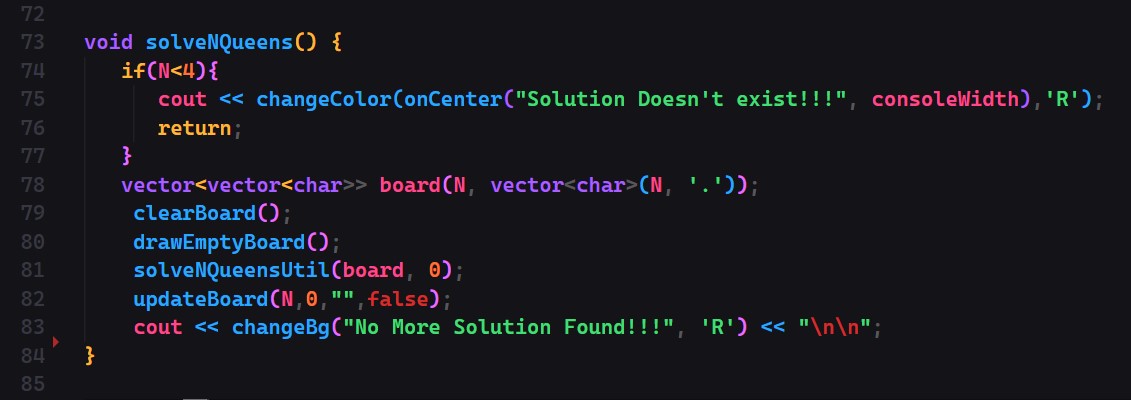
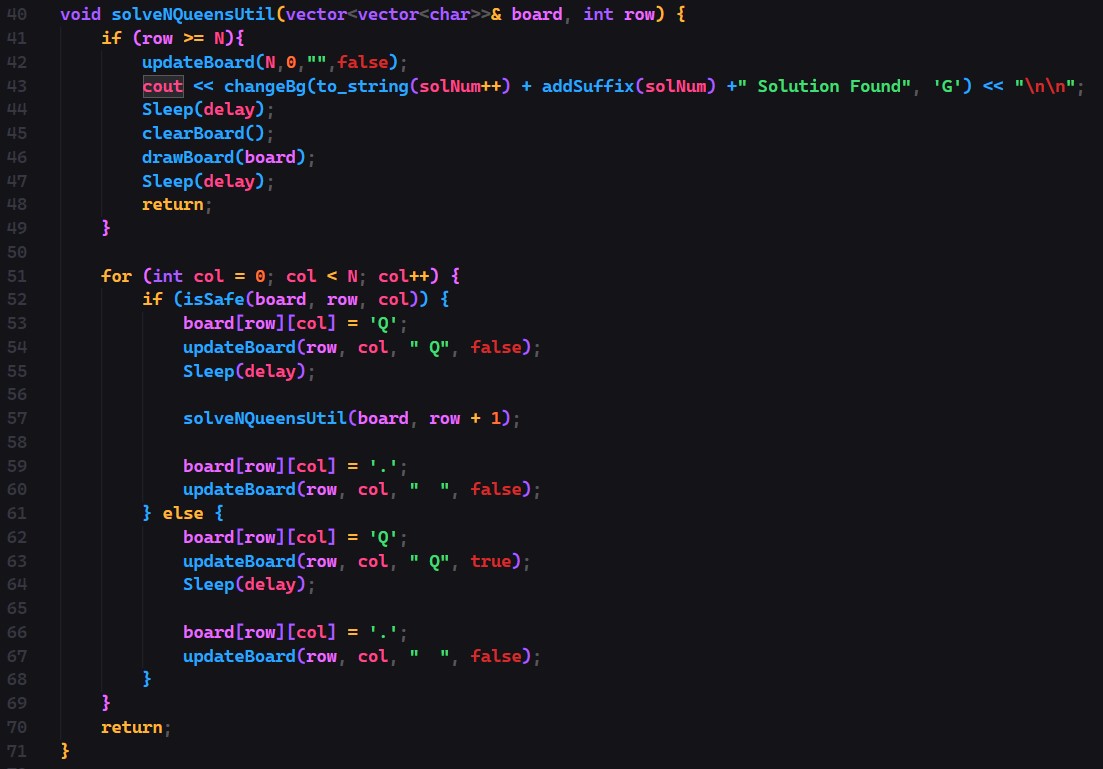
The *visualizerInit* function displays a welcome message and prompts the user to enter the value of N.



## Solver

The solver uses a backtracking algorithm to find a solution to the N-Queens problem.





# Testing

## Test Cases

* **Test Case 1**: N = 4 o Input: 4
  + Expected Output: A valid 4x4 board with 4 queens placed such that no two queens attack each other.
* **Test Case 2**: N = 8 o Input: 8
  + Expected Output: A valid 8x8 board with 8 queens placed such that no two queens attack each other.

## Results

All test cases passed successfully, with the board displaying the queens in the correct positions and with the expected formatting and colors.

# Conclusion

The N-Queens Visualizer in C++ project successfully meets its objectives of solving the N-Queens problem and providing an enhanced console-based visualization. The project demonstrates a strong grasp of algorithmic problem-solving through the implementation of a backtracking algorithm, which efficiently finds solutions for various board sizes. Technically, the project leverages C++ capabilities and Windows API functions to manage console properties and UTF-8 encoding, ensuring broad compatibility, and improving the visual output. The implementation of ANSI escape codes for color output adds a layer of clarity, making the visualization more user-friendly. Additionally, functions like *onCenter* and *changeColor* enhance the user interface, creating a polished and professional application.

## Future Work

* Develop a graphical user interface (GUI) for better visualization.
* Explore more efficient algorithms for solving larger instances of the N-Queens problem.
* Add features to allow users to step through the solution process interactively.

## References

* [N-Queens Problem - Wikipedia](https://en.wikipedia.org/wiki/Eight_queens_puzzle)
* Backtracking - GeeksforGeeks
* [ANSI Escape Codes - Wikipedia](https://en.wikipedia.org/wiki/ANSI_escape_code)