NTUST OOP Midterm Problem Design

Subject: Nonogram

Author: 羅文熠

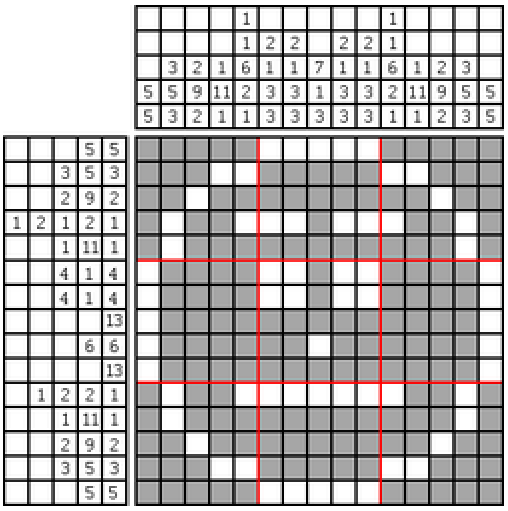
Main testing concept:

Function Basics

■ C++ BASICS	□ SEPARATE COMPILATION AND NAMESPACES
□ FLOW OF CONTROL	□ STREAMS AND FILE I/O
■ FUNCTION BASICS	■ RECURSION
□ PARAMETERS AND OVERLOADING	□ INHERITANCE
□ ARRAYS	□ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ STRUCTURES AND CLASSES	□ TEMPLATES
□ CONSTRUCTORS AND OTHER TOOLS	□ LINKED DATA STRUCTURES
□ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES	□ EXCEPTION HANDLING
□ STRINGS	□ STANDARD TEMPLATE LIBRARY
□ POINTERS AND DYNAMIC ARRAYS	□ PATTERNS AND UML

Description: "This problem is hard to get all points, please make sure you finish other problem first."

Nonograms, also known as Picross or Griddlers, are picture logic puzzles in which cells in a grid must be colored or left blank according to numbers at the side of the grid to reveal a hidden picture. In this problem, you are asked to write a program that solves a given Nonogram puzzle.



Players need to fill in the correct squares in each row and column based on the numbers provided on the grid, in order to reveal a hidden image. The numbers on the grid indicate the number of contiguous squares that need to be filled, with at least one empty square between numbers. By following the number hints, players can gradually deduce which squares need to be filled and which squares do not, until all the squares are filled and the hidden image is revealed.

When solving Nonogram puzzles, players should keep in mind the following:

1. Use logical reasoning based on the numbers provided on the grid to fill in squares.
2. Follow the numbers on the grid and leave at least one empty square between them.

Input：

The input consists of a Nonogram grid with its corresponding row and column hints. The Nonogram grid is represented as a matrix of size N x M, where each cell is either a '#' (indicating a filled cell) or a '.' (indicating an empty cell). The row and column hints are represented as arrays of integers of length N and M, respectively. Each element in the row (column) hint array represents the lengths of the consecutive runs of filled cells in that row (column), **in the order they appear**.

1. Each case begins with a line containing two integers: the number 2 ≤ N ≤ 10 of rows, and the number 2 ≤ M ≤ 10 of columns.
2. Input the hints for each row sequentially, with each row having approximately 0 to M numbers.
3. Input the hints for each column sequentially, with each column having approximately 0 to N numbers.

Output：

If there exists a unique solution to the Nonogram puzzle, output the solved grid with '#' representing filled cells and '.' representing empty cells. If there are multiple solutions, output any one of them. If there is no solution, output "No solution".

Constraints:

2 <= N, M <= 10, the input grid may not have a unique solution. **You may assume that the input Nonogram puzzle is valid**, meaning that the row and column hints are consistent with the grid.

Noted:

The Final(sixth) test case needs you to do a "optimization". If you Try to brutal force, you probably get time limit exceed which is often called(TLE). Assume our modern computer can compute 10^8 times per second, the worst case O(2^(10*10)) >> 2^32(unsigned int size) >> 10^8. So you need to think a better solution!

Input	Output

5 5
1 1
2
3
2 1
2
2
1 3
2 1
2
1
	.#.#.
	..##.
	###..
	##..#
	.##..

5 5
1 1 1
0
0
0
0
0
2 3
0
0
0
	No solution

5 5
1
1 2
2 1
2 1
3
3
4
1
	....#
	.#.##
	##.#.
	##.#.
	###..

3 2	

□ **Eazy,Only basic programming syntax and structure are required.**

□ **Medium,Multiple programming grammars and structures are required.**

■ **Hard,Need to use multiple program structures or more complex data types.**

Expected solving time:

120 minutes

Other notes: