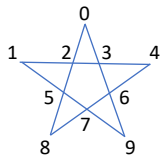
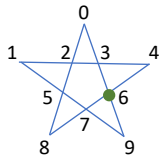
Pentalpha

Pentalpha is a puzzle in which you must place nine stones on the ten intersections numbered 0-9 of a pentagram (i.e., a 5-pointed star).

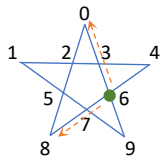


The puzzle is subject to the following rules:

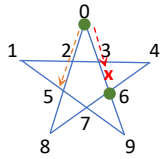
1. Place the first stone on any of the ten intersections.



2. Starting at the location of your first stone X, move in a straight line two intersection points away to Y. You may place a stone on Y only if it is empty.



3. From your second stone, again move in a straight line two intersection points away. You may place a stone at the destination only if it is empty.



4. Repeat until you have placed nine stones.

Write the following function to solve the Pentalpha puzzle.

`bool pentalpha(int place_order[])`
where
`int place_order[]` is an array of 9 elements where you will record the order in which the stones are to be placed (i.e., `place_order[0]` is the first stone, `place_order[1]` is the second stone, etc.).
When your function is called, *the first stone will have already been placed for you* and recorded in `place_order[0]`. You are **not permitted** to change that location. Your task is place the remaining eight stones and record the order of their placements in `place_order[1]` through `place_order[8]`.

Your function should return `true` if you can solve the puzzle based on the first stone's placement, otherwise it should return `false`. Note, however, there is *always* at least one solution (i.e., your function should *always* return `true`) no matter where you start.

File you must submit: `soln_func.cc`