Adding Nodes to the Open Vector

As you've seen from Sebastian's explanation of A* search, the search algorithm keeps a list of potential board cells to search through. In this implementation of A*, we will refer to a board cell along with it's `g` and `h` values as a *node*. In other words, each node will consist of the following values which are needed for the A* algorithm:

- an **x** coordinate,
- a **y** coordinate,
- the **g** value (or *cost*) that has accumulated up to that cell,
- the **h** value for the cell, given by the heuristic function.

In the code, nodes will be implemented with the type `vector<int>`, and should have the form `{x, y, g, h}` for `int`s x, y, g, and h. Also, the open list will be implemented as a C++ vector (of type `vector<vector<int>>`). The goal in this exercise is for you to write a helper function for your A* Search which will add nodes to the open vector and mark them as visited in the grid.

To Complete This Exercise:

1. Write an `AddToOpen` function which accepts the following arguments:
   - Four `int`s, one for each of the `x`, `y`, `g`, and `h` values.
   - References to one `vector<vector<int>>` for the vector of open nodes.
   - Reference to one `vector<vector<State>>` for the grid.
2. The `AddToOpen` function should do two things:
   - Create a `vector<int>` node with the form `{x, y, g, h}` and push the node to the back of the open vector.
   - Set the grid value for the `x` and `y` coordinates to the enum value `kClosed`. We have added `kClosed` to the set of enum values.

main.cpp

```
1 #include <fstream>
```