

Please put your answer to question 5 here:

```
Vector<string> allPaths(Grid<bool> &g) {
    return allPaths(g, 0, 0, "");
}

// helper function:
Vector<string> allPaths(Grid<bool> &g, int r, int c,
                        string path) {
    Vector<string> paths;
    if (!g.inBounds(r,c) or g[r][c]) {
        // already visited, or out of bounds
        return paths;
    }

    if (r == g.numRows() - 1 and c == g.numCols() - 1) {
        // we've reached the end
        paths.add(path);
        return paths;
    }

    // general idea: mark current position as visited,
    // then visit all neighbors
    g[r][c] = true; // visited
    paths += allPaths(g, r-1, c, path + "n");
    paths += allPaths(g, r, c+1, path + "e");
    paths += allPaths(g, r+1, c, path + "s");
    paths += allPaths(g, r, c-1, path + "w");
    // unmark
    g[r][c] = false;
    return paths;
}
```