```cpp
#include "Definitions.h"
#include "PostProcessorVtk.h"


int main() {

  // Geometrical properties
  // TODO: Set positive integer for numberOfElementsX and numberOfElementsY
  int numberOfElementsX = 10;
  int numberOfElementsY = 10;

  cout << "Let's build a mesh with numberOfElementsX=" << numberOfElementsX << " and numberOfElementsY=" << numberOfElementsY << endl;
  cout << "This should give a grant total of " << numberOfElementsX*numberOfElementsY*2 << " elements" << endl;

  // TODO: Set positive side lengths
  double sideLengthX = 10.0;
  double sideLengthY = 10.0;

  cout << "The side lengths are: sideLengthX=" << sideLengthX << ", sideLengthY=" << sideLengthY << endl;

  // Initialization of mesh
  Mesh rectangularTriangleMesh;

  // Incremental side length of one element
  // TODO: based on numberOfElementsX,numberOfElementsY
  //       and sideLengthX,sideLengthY, define dx,dy
  double dx = sideLengthX / numberOfElementsX;
  double dy = sideLengthY / numberOfElementsY;

  cout << "The incremental side lengths are: dx=" << dx << ", dy=" << dy << endl;




  // TODO: Add all nodes into the public member _nodes of rectangularTriangleMesh.
  //       You may want to use a for-loop or even a nested-for-loop to achieve this.
  // HELP: This is how you create three nodes and it onto the aforementioned vector
  //       (Reminder: 'nested' means a for-loop inside a for-loop).
  Vector2d nodeLocation;
  for (double y = 0.0; y <= sideLengthY; y += dy) {
    for (double x = 0.0; x <= sideLengthX; x +=dx){
        nodeLocation(0) = x;
        nodeLocation(1) = y;
        Node node(int((numberOfElementsX+1)*(y/dy)+ x/dx), nodeLocation);
        rectangularTriangleMesh._nodes.push_back(node);
    }
  }




  // TODO: The goal of the next section is to create a triangular element as shown in the
  //       assignment sheet and to add it onto the _connectivity public member variable of
  //       rectangularTriangleMesh.
  //       In order to build the entire mesh, this then has to be repeated for all elements, the total
  //       number of which is determined via numberOfElementsX and numberOfElementsY. You may want to use a
  //       nested for-loop
  // HINT: See your job similar to the one of a floor tiler, but instead of explicitely saying where every
  //       single tile goes, you tell the program how to place one or two tiles and then let it repeat this
  //       process numberOfElementsX times in the x-direction and again repeat this numberOfElementsY times
  //       in the y-direction.
```

```
64
65      // HELP: This is how we create ONE triangular element:
66    // elementNumber = numberOfElementsX*numberOfElementsY*2
67      for (double i = 0.0; i < sideLengthX; i +=dx){
68          for (double j = 0.0; j < sideLengthY; j+=dy){
69              array<int, 3> connection1;  // this holds the node IDs of the three nodes
                of the element
70              array<int, 3> connection2;
71              connection1[0] = int ((numberOfElementsX+1)*(j/dy)+i/dx);
72              connection1[1] = int ((numberOfElementsX+1)*(j/dy)+i/dx+1);
73              connection1[2] = int ((numberOfElementsX+1)*(j/dy+1)+i/dx+1);
74              connection2[0] = int ((numberOfElementsX+1)*(j/dy)+i/dx);
75              connection2[1] = int ((numberOfElementsX+1)*(j/dy+1)+i/dx+1);
76              connection2[2] = int ((numberOfElementsX+1)*(j/dy+1)+i/dx);
77              // HELP: This is how you add it onto the _connectivity member of
                rectangularTriangleMesh
78              rectangularTriangleMesh._connectivity.push_back(connection1);
79              rectangularTriangleMesh._connectivity.push_back(connection2);
80          }
81
82      }
83      // Make VTK file
84      PostProcessors::Vtk::makeUndeformedMeshFile(rectangularTriangleMesh,string("Mesh"));
85
86      cout << "Succesfully ran main and created .vtu file" << endl;
87
88      return 0;
89    }
90
91
92
93
```