```
void Application::tree(int tree_subdiv, float tree_height, float tree_top_radius, float tree_bottom_radius,   // tree parameters
    int branch_subdiv, int num_branch_rows, float branches_start, float branches_end, float branch_top_radius, float branch_bottom_radius,  // branch parameters
    float uniform_scale, float tree_x_position, float tree_y_position, float tree_z_position, float tree_x_rotation, float tree_y_rotation, float tree_z_rotation) {   // universal parameters

        // draw tree trunk
    bm_tree_object_ptr = new basic_model;
    bm_tree_object_ptr->mesh = cylinder(tree_subdiv, tree_top_radius, tree_bottom_radius, tree_height, true, true, tree_x_position,
                                        tree_y_position, tree_z_position, tree_x_rotation, tree_y_rotation, tree_z_rotation);

    bm_tree_object_ptr->color = vec3(0.45, 0.38, 0.3);    // assign color to object
    bm_tree_object_ptr->shader = just_shader;
    m_tree_objects.push_back(bm_tree_object_ptr);      // append current basic_model object

    // compute branch variables
    float branches_height = branches_start - branches_end;
    float branch_height_incr = branches_height / num_branch_rows;
    float branch_radius_incr = (branch_bottom_radius - branch_top_radius) / num_branch_rows;
    float tree_top = tree_height * 0.5;
    float tree_bottom = -tree_height * 0.5;

    // draw tree branches
    for (int i = 0; i < num_branch_rows; i++) {
        bm_tree_object_ptr = new basic_model;

        bm_tree_object_ptr->mesh = cylinder(branch_subdiv, 0, branch_top_radius + (branch_radius_incr * i), branch_height_incr, true, true,
            tree_x_position, tree_y_position + tree_top + branch_height_incr - (branch_height_incr * i), tree_z_position, tree_x_rotation, tree_y_rotation, tree_z_rotation);
        bm_tree_object_ptr->color = vec3(0.1, 0.7, 0);   // assign color to object
        bm_tree_object_ptr->shader = just_shader;
        m_tree_objects.push_back(bm_tree_object_ptr);   // append current basic_model object
    }

}
```