1. Consider a random network with, say, 7 nodes. What is the probability that every node has exactly 3 neighbors?

2. What can you say about the sign of the expression $(2u+1)\ln\left(1+\frac{1}{u}\right)-2$ for $u>0$?

3. In some city a car starts moving from the central square to some direction. The driver applies a funny algorithm turning right at every intersection. More precisely, the car turns to the nearest right if many available. Assume that all roads are two-ways (in both directions) and the number of roads at any intersection is greater than 2 and can be more than 4. Besides that, there can be multilevel roads so that you cannot assume that roads over the same x;y intersect. Can you guarantee that the driver will return back to the central square in every city like this?

4. The definition of a class below is rich in errors and style issues.

```
class Complex
{
public:
        Complex( double real, double imaginary = 0 )
                : _real( real ), _imaginary( imaginary )
        {
        }
        void operator + ( Complex other )
        {
                _real = _real + other._real;
                _imaginary = _imaginary + other._imaginary;
        }
        void operator << ( ostream os )
        {
                os << "(" << _real << "," << _imaginary << ")";
        }
        Complex operator ++ ()
        {
                ++_real;
                return *this;
        }
        Complex operator ++ ( int )
        {
                Complex temp = *this;
                ++_real;
                return temp;
        }
private:
        double _real, _imaginary;
    };
```

Please indicate the issues you see. Fix those issues and rewrite the class definition.

5. **Checking whether a tree is balanced.** A binary search tree is an efficient data structure that allows fast searching, insertion and deletion of an element with specific key. These advantages are maximal when the tree is balanced. As an extreme bad case we can have a "tree" degenerated to just a linked list. In this case all good qualities aren't available. You need to implement a function checking whether the given tree is balanced or not. To be precise, we call the tree balanced if for each node the heights of its left and right sub-trees differ by at most 1. Please provide a generation of a tree at random and a test function that generates trees and then checks whether those trees are balanced with different answers happening in the test.

6. Military tanks ride only by asphalt roads on the planet BARSOOM. While moving, tanks destroy roadbed under themselves. A very modern tank **Armata** made a trip from the CAPITAL to ZODANGA city. We know that minimal road width on the BARSOOM didn't change after this trip. What values can **Armata**'s width take? The width of every road is known. We expect that you provide a C++ code for the solution of this problem. Also we expect algorithm explanation (with prove of correctness and complexity estimation). You should take into account that BARSOOM planet can have a rich number of roads.

   The format of data file with roadmap is up to you. You don't have to use any libraries except for input/output, random and sorting. Roads width are real numbers; number of cities <=10^6; number of roads <=10^6.
   Please provide a generation of input samples with any number of cities and roads in your code (choice of CAPITAL and ZODANGA, roads' placement, their capacities and width should be generated at random). The code should also contain a test function, which reads the data file, calls your solution method, and returns the result. The result of compilation should be console application (e.g. opt.exe).

| Console input | What does it mean |
|---|---|
| opt.exe -g 20 100 bars.txt | Generate graph with 20 nodes and 100 roads and write to bars.txt<br>*Note: if number of roads >= number of nodes then graph should connected* |
| opt.exe -t mars.txt | Test graph from mars.txt |