- 👋 Hi, I’m basanta
- 👀 I’m interested in c++ and embedded, 5G, IoT
- 🌱 I’m currently learning c++ and data structures
- 🤝 I’m looking to collaborate on open source projects on c++, linux
- 📫 How to reach me: Mail at basanta229@gmail.com, Message at https://www.linkedin.com/in/bbasanta/

<!---
basantsansa/basantsansa is a ✨ special ✨ repository because its `README.md` (this file) appears on your GitHub profile.
You can click the Preview link to take a look at your changes.
--->

This branch holds c++ implementation of below problem statement -

Depth Scanner
---
A group of scientists has designed an imaging gadget that is able to measure the depth of materials over extremely long distances. As an initial test run, the scientists decide to create a 3D model of an area located on the Moon.
The imaging gadget, or "depth scanner" as they call it, measures depth readings in a target area. The final output of a single scan is a M-by-N pixel depth image that can be used to draw a 3D image of the measured target. Each pixel corresponds to the average depth in a 1-by-1 meter area. The depth scanner operates such that it measures the target one row at a time and, after each measurement, it outputs N integer values. After M successive measurements, the full scan is complete. Note that the depth scanner doesn't have an internal buffer to store the measurements, so it can only output data after each row measurement.
Your task is to create a C++ class that can be used to store the measurements of the depth scanner. Additionally, the group of scientists have identified two primary use cases for the depth scanner: 3D visualization and smoothness estimation.
Visualization happens such that a threshold is given and if the pixel value is greater than the threshold, then '#' should be printed. Otherwise, '.' should be printed. The result is a M-by-N grid of '#' and '.', where N is the row width and M is the number of rows that have been measured. By layering the images, a 3D construction of the measured target can be created.
The smoothness of the target surface is the total number of smooth locations. A location is defined to be smooth if the depth reading at that location multiplied by the number of neighbours differs from the sum of readings of the neighbouring locations by no more than 10 depth units. For example, if the depth reading at location P is 3 depth units, P has 5 neighbours and the sum of their depths is 25, then P is smooth as 25 - 3 * 5 = 10. If the depth at P was 8, P wouldn't be smooth as 25 - 8 * 5 = -15 < -10. As a final example, the smoothness of the area in the following scan is 7:
1 7 -2 0
-4 1 1 0
3 1 1 -4
0 16 9 0
Implement both the visualization and smoothness estimation as defined above. Note that each location has either 3, 5 or 8 neighbours.
To test your implementation, measurement data is provided. The first line of the file defines the number of rows N and columns M in the scan, separated by a

single space. After this, there are N rows of M integer values separated by a single space.

**Source code files**: node.h depthScanner.h depthScanner.cpp
**Test input data files**: in2x2.txt, in3x3.txt, in4x4.txt, in5x5.txt, in50x50.txt

**Compilation**:
Using a small script 'gccp' with below content.
#/bin/sh
echo compiling C++ using -ansi -std=c++11 -pedantic-errors -Wall
g++ -ansi -std=c++11 -pedantic-errors -Wall $1 $2 $3

So shell command executed: gccp node.h depthScanner.h depthScanner.cpp

**Execution**:
example command with input data file passed on command-line :
./a.out in50x50.txt