```cpp
#include <random>
#include <array>
#include <cmath>
#include <fstream>

//The sample size for plotting final distribution - this many numbers will be drawn
constexpr size_t samplesize = 10000;

int main()
{
    std::array<double, samplesize> Z{}; //array to store the values, in case we need

    std::random_device dev; //Responsible for getting a random seed from OS
    std::mt19937_64 rng(dev());    //Mersenne Twister engine with the seed for generating pseudo-random numbers
    std::uniform_real_distribution<double> dist(0,1); // distribution in range [0, 1]

    double sigmainverse = sqrt(12.0);   // 1/(standard deviation) for the uniform distribution
    double mean = 0.5;  //mean of the uniform distribution

    std::ofstream outfile;  //file handle to save the results in a file
    outfile.open("./output/problem1.txt", std::ios::out | std::ios::trunc);

    for(auto& Zi : Z){  //Loop through the array to store the values
        Zi = sigmainverse * (dist(rng) - mean);    //   calculate Y1 and store in the array
        outfile << Zi << std::endl; //write to the output file
    }
    outfile.close();    //when done, close the file.
}
```