```
using Distributions;
using StatsBase;

N = [50, 250, 750, 1500, 3000];
M = [100, 1000];
x = 0.5; a = -5; b = 5;
for n in N
    u_grid = collect(range(0, length=n+2, stop=1)[2:n+1]);
    a_grid = Array{Float64, 1}(undef, n);
    b_grid = Array{Float64, 1}(undef, n);
    unnormalised_posterior_a = Array{Float64, 1}(undef, n);
    unnormalised_posterior_b = Array{Float64, 1}(undef, n);
    unnormalised_posterior = Array{Float64, 1}(undef, n);
    expectation = Array{Float64, 1}(undef, n);
    for i in 1:1:n
        u = u_grid[i];
        a_grid[i] = x - sqrt(-2*log(u));
        b_grid[i] = x + sqrt(-2*log(u));
        unnormalised_posterior_a[i] = 1 / sqrt(-2*log(u)) / ((x-sqrt(-2*log(u)))^2
+ 1) / n;
        unnormalised_posterior_b[i] = 1 / sqrt(-2*log(u)) / ((x+sqrt(-2*log(u)))^2
+ 1) / n;
        unnormalised_posterior[i] = unnormalised_posterior_a[i] +
unnormalised_posterior_b[i]
    end
    A = sum(unnormalised_posterior);
    expectation = a_grid .* unnormalised_posterior_a + b_grid .*
unnormalised_posterior_b;
    println("-------------------- n = ", n, " --------------------");
    println("With transforming: E[Y|X=0.5] = ", sum(expectation)/A)

    for m in M
        sample_index = sample(1:n, m, replace = true);
        samples = expectation[sample_index];
        sample_expectation = sum(samples) * n / m / A;
        println("m = ", m, ": ", sample_expectation);
    end

    if n <= 1000
        y_grid = collect(range(a, length=n, stop=b));
        newa = a;
    elseif n > 1000 && n <= 2000
        nm = 1000;
        na = round(Int, (n-nm)/2);
        l = (b-a)/(nm-1);
        newa = a - l*na;
        y_grid = collect(range(newa, step=l, length=n));
        y_grid = collect(range(newa, step=l, length=n));
    elseif n > 2000
        nm = round(Int, n/2);
        na = round(Int, (n-nm)/2);
        l = (b-a)/(nm-1);
        newa = a - l*na;
        y_grid = collect(range(newa, step=l, length=n));
```

```
        end
        for i in 1:1:n
            y = y_grid[i];
            unnormalised_posterior[i] = (b - newa) * exp(-(x-y)^2 / 2) / (1 + y^2) / n;
        end
        A = sum(unnormalised_posterior);
        posterior = unnormalised_posterior / A;
        expectation = y_grid .* posterior;
        println("Without transforming: E[Y|X=0.5] = ", sum(expectation));
        println()
    end
```

Result

---------- n = 50 ----------

With transforming: E[Y|X=0.5] = 0.22622987066654882

m = 100: 0.3682260651367699

m = 1000: 0.2317674091333775

Without transforming: E[Y|X=0.5] = 0.2661755641372646

---------- n = 250 ----------

With transforming: E[Y|X=0.5] = 0.24940741511377373

m = 100: 0.30822164931882245

m = 1000: 0.2582741085306477

Without transforming: E[Y|X=0.5] = 0.26617525294322397

---------- n = 750 ----------

With transforming: E[Y|X=0.5] = 0.25673216025090073

m = 100: 0.19320046654301248

m = 1000: 0.2969050113698817

Without transforming: E[Y|X=0.5] = 0.26617519291542185

---------- n = 1500 ----------

With transforming: E[Y|X=0.5] = 0.2595673372913161

m = 100: 0.3896443492499317

m = 1000: 0.2756103506632583

Without transforming: E[Y|X=0.5] = 0.26617612339045515

---------- n = 3000 ----------

With transforming: E[Y|X=0.5] = 0.26153801725366294

m = 100: 0.29191283535796847

m = 1000: 0.2558704635757391

Without transforming: E[Y|X=0.5] = 0.2661761233906986