```cpp
#include <bits/stdc++.h>
//Cairo University Faculty of Engineering Computer Engineering Department
//Numerical Analysis Fall 2018 semester MTH 325
//name:Yahia Ali Sec:2 BN:32
//problem 3.17
//this code for solve Integration (ln(x)*ln(y/M))/(sqrt(x*y/M)) dxdy, x from start to
end and y from M*start to M*end
//using Simpson's 1/3 formula with n=N where N is even number
double const infinte= 99999999.0; //define infinte for double numbers
double const zero=0.0000001; //define zero for double numbers
double func(double x) //calculate ln(x)/sqrt(x)
{
    if(x<=zero) //stop if input causes error (x is negative or zero)
        return infinte;
    return log(x)/sqrt(x);
}
double solve(double start,double end, int N,double M)
{
    if(N&1 || N<1) //if N is odd or negative stop
        return infinte;
    double result=0.0; //final result will store here
    double sumeven=0.0; // summation of all f(x) with even index
    double sumodd=0.0; // summation of all f(x) with odd index, first and last index
    (1,N+1) didn't include
    double h=(end-start)/double(N); //calculate h
    double x=start+h; //store current x
    for(int index=2;index<=N;index++) //start from index 2 to index N
    {
        if(index&1) //if index is odd add f(x) to sumodd
            sumodd+=func(x);
        else //else add f(x) to sumeven
            sumeven+=func(x);
        x+=h; //increase x by h
    }
    result=func(start)+(4.0*sumeven)+(2.0*sumodd)+func(end);
    result=result*h/3.0; // now result contain Integration = (ln(x)/sqrt(x))dx from
    start to end >> I simp1/3
    result= (result*result)*M; //calculate 2* (I simp1/3)^2
    return result;
}
int main()
{
    //test for start=1,end=2,n=8 and M=2
    printf("%.8f",solve(1.0,2.0,8,2.0)); //print the result 8 digit after Fraction
    return 0;
}
```