% Simpson's 1/3 Rule

syms x; % Declare symbolic variable

fun = input('Enter the function f(x) (e.g., x^2+2\*x): ', 's'); % Input function as a string

f = str2func(['@(x) ' fun]); % Convert the input string to a function handle

% Input integration limits and number of partitions

x\_1 = input('Enter the lower limit a: '); % Lower limit of integration

x\_n = input('Enter the upper limit b: '); % Upper limit of integration

n = input('Enter the number of partitions n (must be even): '); % Number of partitions

% Verify if the number of intervals is even

if mod(n, 2) ~= 0

disp("Simpson's 1/3 Rule is not applicable (requires number of intervals to be even).");

else

% Calculate step size

h = (x\_n - x\_1) / n;

% Generate x and y values

x\_values = x\_1:h:x\_n; % Generate x values

y\_values = arrayfun(f, x\_values); % Evaluate f(x) at each x value

% Apply Simpson's 1/3 Rule

A = y\_values(1) + y\_values(end); % Add first and last terms

A = A + 4 \* sum(y\_values(2:2:end-1)); % Add 4 times odd-indexed terms

A = A + 2 \* sum(y\_values(3:2:end-2)); % Add 2 times even-indexed terms

A = (h / 3) \* A; % Final area computation

% Display the result

fprintf('The area under the curve using Simpson''s 1/3 Rule is: %.6f\n', A);

end