Make a numerical integrator:

The simplest integrator uses Riemann sums.
http://people.hofstra.edu/Stefan_Waner/realworld/integral/numint.html

Don’t use any built-in functions, like **sum( )**.

In main.m:

```
clear;clc;
% numerically integrate y(x) = ax^2 + bx + c from lowerlimit to upperlimit
% y(x) = x^2 + 2x + 1
a = 1;
b = 2;
c = 1;

% from 0 to 1
lowerlimit = 0;
upperlimit = 1;

% number of discretizations
N = 100;

fprintf('The integral of \n')
fprintf('%6.2fx + %6.2fx + %6.2f \n', a,b,c)
fprintf('from %6.2f to %6.2f \nis %6.2f \n' , ...
lowerlimit,upperlimit,integ(a,b,c,lowerlimit,upperlimit,N) )
```

In integ.m:

```
function [ totarea ] = integ(a,b,c,xstart,xend,N)
% This fn calculates the integral of ax^2 + bx^2 + c from xstart to xend
% Using Riemann summation

totarea = 0;
dx = (xend-xstart)/N;
x=xstart;
for i=1:N
  y = a*x^2 + b*x + c;
  totarea = totarea + y*dx;
  x = x + dx;
end

endfunction
```

Another way of writing the user-defined function using the function **linspace( )**:

```
function [ totarea ] = integ(a,b,c,xstart,xend,N)
totarea = 0;
dx = (xend-xstart)/N;
x = linspace(xstart,xend,N);  % creates array of linearly spaced values
for i=1:N
   y = a*x(i)^2 + b*x(i) + c;
   totarea = totarea + y*dx;
end

endfunction
```