2) function ~~f(x)~~ x = f(n)

```
x=1;
for i=1:n
  for j=1:n
    x=x+1;
```

1) Find the runtime of the algorithm mathematically

Inner loop:

It executes n times for each phase in outer loop.

∴ n executions for inner loop

Outer loop:

It executes n times from i=1 to i=n

∴ n executions for outer loop

Total executions = $\sum_{i=1}^{n} \sum_{j=1}^{n} 1$

$= \sum_{i=1}^{n} n$

$= n \times n$

$= n^2$

∴ The total runtime of the algorithm is $O(n^2)$

3) Find polynomials that are upper and lower bounds on your curve.. From this specify big-O, big-Omega, big-theta is

Big-O Notation (Upper bond)

$O(n^2)$ - function does not grow faster than quadratic

Big-Omega

$\Omega(n^2)$ - function does not grow slower than quadrat

Big Theta

$\Theta(n^2)$ - function grows asymptotically as n^2)

If I modified the function to be:

```
x=f(n)
x=1;
y=1;
for i=1:n
    for j=1:n
        x=x+1;
        y=y+i+j;
```

4) Will this increase how long it takes the algorithm to run?

$y=i+j$, actual time taken by the modified function will be slightly longer due to it.

$\therefore O(n^2)$ is the time complexity.

5) Will it effect your results from first function?

It doesn't effect much in the time complexity.