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

1. Find the runtime of the algorithm mathematically (I should see summations).

⇨ To analyze the runtime of the given function mathematically, let's express the number of operations using summations.

- The initialization of x=1 takes **constant time: O(1)**.
- The **outer loop** runs $from\ i = 1to\ i = i = n$, meaning it executes **n times**.
- The **inner loop** runs from j=1$to\ j = n$, meaning it also executes **n times** for each ii.
  - The statement x=x+1 inside the inner loop is executed once per inner loop iteration.

Step 2: Writing the Summation

The number of times the innermost operation executes can be expressed as:

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

⇨ Since the inner sum runs n times for each of the n iterations of the outer loop, we expand:

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

$$\sum_{i=1}^{n} n =. n = n^2$$

⇨ Thus, the total number of operation:

$\text{T}(n) = \Theta(n^\wedge 2)$

Final Answer:

The runtime complexity of the function is: (n^2)