Seeminglymeaningless's Evil Problem

Ah... don't you love it when algebra teachers give you calculus problems?

So, basically, I am going to make my computer program do all the calculus for you. Because that's what computers do. I am also going to be a horrible person and complete this problem for you, with the help of this computer program. Why? Because your teacher is going way beyond your level by assigning you a calculus problem, and that's not really fair. Still, I explained it step by step, so please study it carefully, just in case this problem ever comes up again.

First, we need to find out the derivative of this bad boy, right? So here it is, using handy dandy software! :) It says Log(x)... that's just the software's way of saying natural log. Because only chemists use regular log. And they don't need math! :P

So, let's start. Our function is:

```
x * Log[x] - 6
```

We need to find the first derivative of it! And, since you're in algebra and you pretty much don't want to bother with calculus at this point, we plug it in the software and make it do it for us! :D

```
In[18]:= D[x * Log[x] - 6, x]

Out[18]= 1 + Log@xD
```

As a side note, you can probably do this if you have a TI-89 calculator. Ask me if you want to know.

Now, Newton's method!

Newton's method, on this software, will look a little like this:

```
x1 = xo - f (xo) / f ' (xo)
```

...except with all the pretty functions and everything. Then, once we find x1, we put it where xo is and find x2! On and on we go until it finally seems to settle at just ONE point.

So... next question is, what is xo? The answer? It depends. Normally, we would start with zero, because zero is usually a good number to start with! However, because we are are dealing with log functions, this gets a little bit trickier. Let's start with 0.5. Because that's a good place to start!

```
In[10]:= x1 = xo - (xo * Log[xo] - 6) / (1 + Log[xo]) /. xo -> 0.5

Out[10]= 21.1828
```

Note that the back part of this equation (the /.xo->0.5 part) is nothing to be freaked out about. It just means that we're replacing xo with 0.5.

Now, do this again!

```
In[12]:= x2 = x1 - (x1 * Log[x1] - 6) / (1 + Log[x1])

Out[12]= 6.70652
```

And again...

In[13]:=
```
x3 = x2 - (x2 * Log[x2] - 6) / (1 + Log[x2])
```

Out[13]=
```
4.37691
```

...and again.

In[14]:=
```
x4 = x3 - (x3 * Log[x3] - 6) / (1 + Log[x3])
```

Out[14]=
```
4.19042
```

In[15]:=
```
x5 = x4 - (x4 * Log[x4] - 6) / (1 + Log[x4])
```

Out[15]=
```
4.18876
```

In[16]:=
```
x6 = x5 - (x5 * Log[x5] - 6) / (1 + Log[x5])
```

Out[16]=
```
4.18876
```

Two same numbers in a row! Looking good!

In[17]:=
```
x7 = x6 - (x6 * Log[x6] - 6) / (1 + Log[x6])
```

Out[17]=
```
4.18876
```

And this is the same number again! Definitely good. If I look at the full number, it's:

```
4.1887601193156`
```

And there you go! We can assume that this is a pretty good approximation!

Mind you, it's just that... an approximation. But it's still fairly accurate, as no doubt Wikipedia has informed you on. ;)