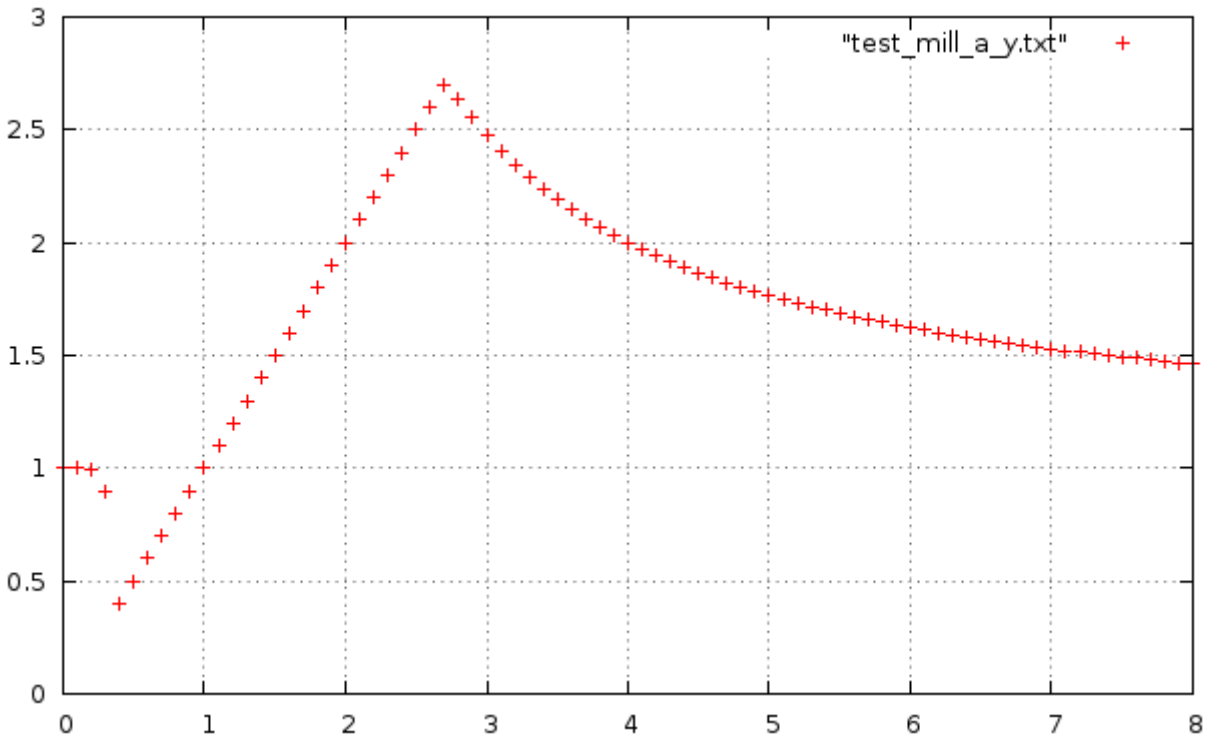
The setup:

```
#include <iostream>
#include <cmath>
#include <fstream>

using namespace std;


double tetration (double x, long order)    //tetration = powertower in this document
{
   double y = pow(x,x);

   for( long i = 0; i < order; i++)
   {
      y = pow(x,y);
   }

   return y;
}



int main()
{
     fstream fs;
     fs.open ("test_mill_a_y.txt", fstream::in | fstream::out | fstream::app);

   for(double a = 0.0; a <= 8.0; a += 0.1)
   {
      double x = pow(a,1.0/a);
      double y = tetration(x,1000000);
      double z = exp(log(y)/y);

      cout << a << "\t" << y << endl;
      fs << a << "\t" << y << endl;
   }
   fs.close();
   return 0;
}
```

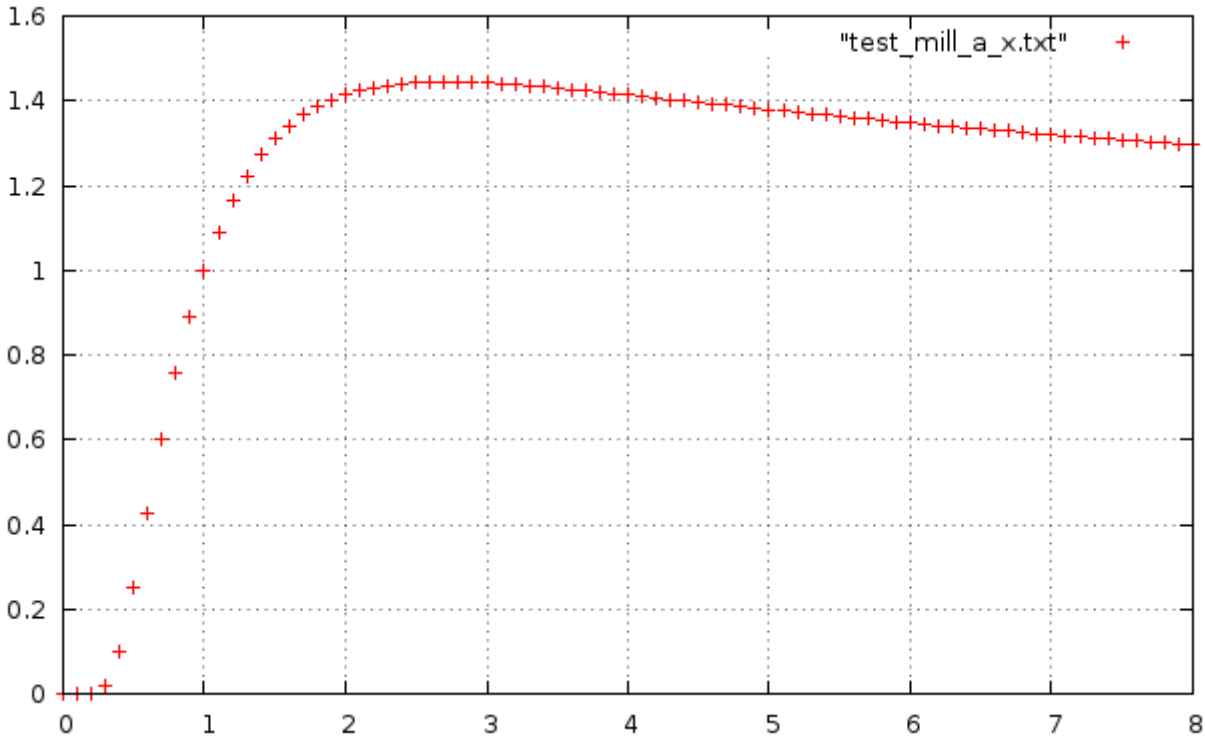
Experiment:(the challenge)



f(a):
X-axis: 0 <= a <= 8
Y-axis: powertower(x) with x = a ** (1/a) @1 000 000 Iterations.

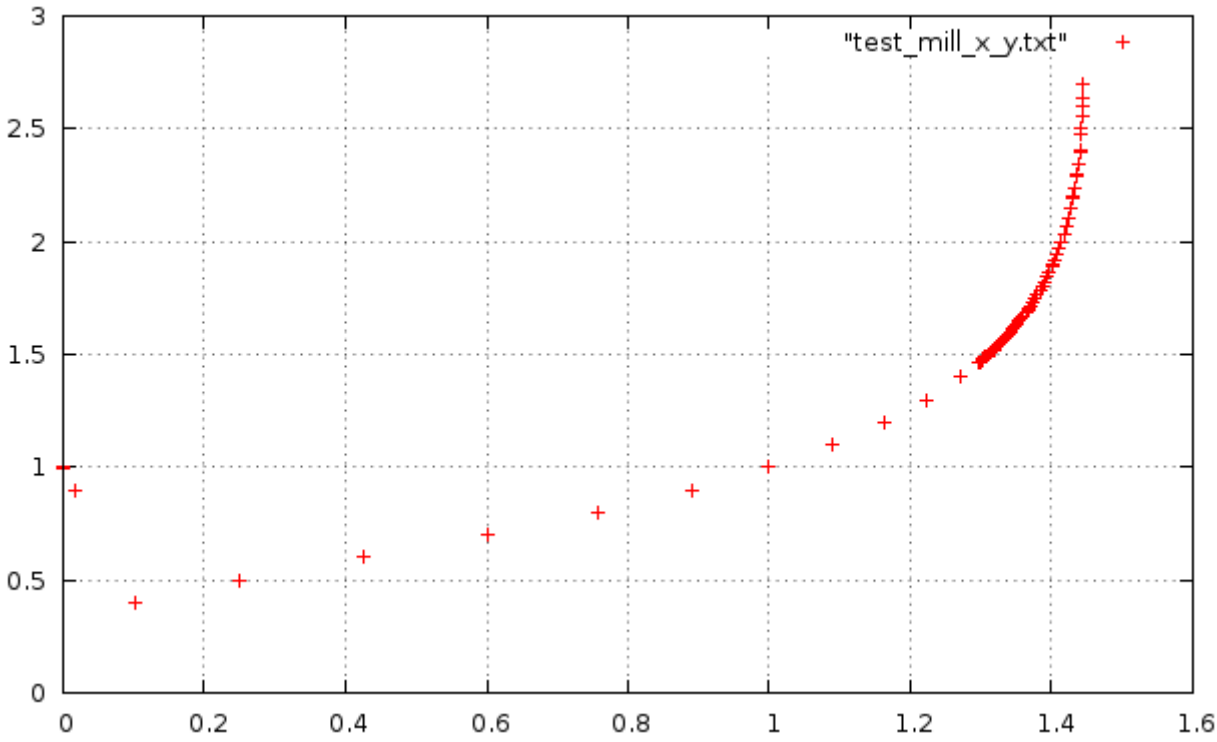
For 1/e <= a <= e: f(a) looks linear.
But why?

Let us see, how x depends from a:



x-axis: 0 <= a <= 8
y-axis: a**(1/a) ; the maximum is at a = e

x versus powertower-function



x-axis: x= a **(1/a) with 0 <= a <= 8
y-axis: powertower(x) @1 000 000 Iterations.