Euler Problem #92

This is the function that follows the rule of the number chain

In[22]:=
```
chain[initialNum_] :=
 Module[{digits, squaredDigits},
  (*Get each of the digits of the initial number input*)
  digits = IntegerDigits[initialNum];
  (*The square of each digit*)
  squaredDigits = #^2 & /@ digits;
  (*The sum of the squared digits*)
  Total[squaredDigits]
 ]
```

Example of this function working (64+81=145)

In[69]:=
```
chain[89]
```

Out[69]= 145

This is a function that repeatedly applies the function until the last term becomes either 1 or 89 (It's given by the problem that all numbers end at 1 or 89)

In[181]:=
```
untilChainStops[initialNum_] := NestWhile[chain[#] &, initialNum, # ≠ 89 && # ≠ 1 &]
```

In[185]:=
```
showtheChain[initialNum_] := NestWhileList[chain[#] &, initialNum, # ≠ 89 && # ≠ 1 &]
```

Examples of this function working/ Showing the chain itself (this is just for you to show the process!)

In[191]:=
```
untilChainStops[44]
```

Out[191]= 1

In[192]:=
```
showtheChain[44]
```

Out[192]= {44, 32, 13, 10, 1}

In[187]:=
```
untilChainStops[145]
```
Out[187]= 89

In[188]:=
```
showtheChain[145]
```
Out[188]= {145, 42, 20, 4, 16, 37, 58, 89}

In[197]:=
```
showtheChain[9999]
```
Out[197]= {9999, 324, 29, 85, 89}

Test every number until a hundred thousand (The Original Problem requires for ten million, but that computation took WAY too long...)

In[195]:=
```
results = untilChainStops[#] & /@ Range[100 000];
```

How many numbers end at 89?

In[196]:=
```
Length@Select[results, # == 89 &]
```
Out[196]= 85 623