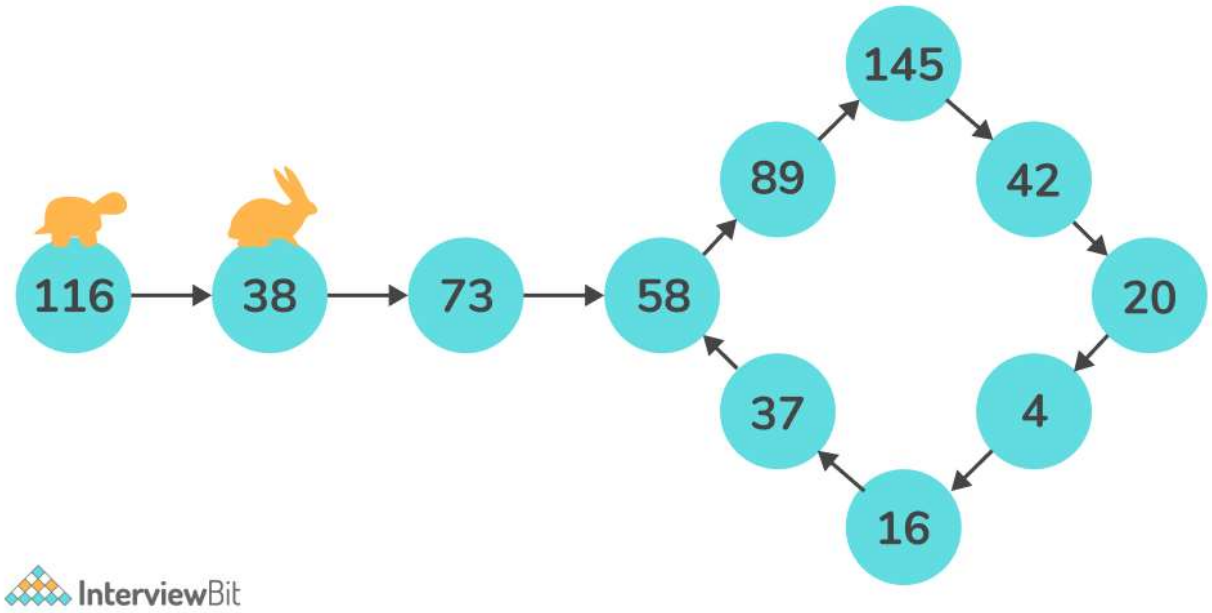
Approach 2: Floyd's Cycle Detection Algorithm

If an integer is not a Happy number, after certain steps it gets into the cycle. Now, this cycle can be thought of as a linked list and then our task would be to simply detect if a cycle exists in a linked list.



Floyd's Cycle Detection Algorithm is based on two-pointers- **fast** and **slow** pointers. The slow pointer moves ahead by one step at a time, whereas the **fast** pointer moves ahead by two steps at a time. In case, there is a cycle, both the **fast** and **slow** pointer eventually meet after at a certain point.

Algorithm:

- Initialise slow pointer with **N** and fast pointer with the sum of squares of digits of **N**.
- While **fast pointer** doesn't become equal to **1** and **slow** and **fast** pointer doesn't meet,
    - Replace **slow** with the sum of squares of digits of **slow.**
    - Replace **fast** with the sum of squares of digits of **fast.**
- Return **True** if **fast** becomes 1, else return **False.**