Problem Description: Happy Number Check

Objective: Write a function `isHappyNumber(n)` that takes a positive integer `n` and returns `True` if `n` is a happy number, and `False` otherwise.

Definition: A happy number is a number which eventually reaches 1 when replaced by the sum of the square of each digit. If it falls into a cycle that does not include 1, it is not a happy number. For example, 19 is a happy number because:

1. $1^2 + 9^2 = 1 + 81 = 82$
2. $8^2 + 2^2 = 64 + 4 = 68$
3. $6^2 + 8^2 = 36 + 64 = 100$
4. $1^2 + 0^2 + 0^2 = 1$

Parameters:

- `n (int)`: A positive integer which needs to be checked if it is a happy number.

Returns:

- `bool`: `True` if `n` is a happy number, `False` otherwise.

Examples:

1. **Example 1:**
    - **Input:** `n = 19`
    - **Output:** `True`
    - **Explanation:** The number 19 eventually reaches 1 through the process described above.
2. **Example 2:**
    - **Input:** `n = 2`
    - **Output:** `False`
    - **Explanation:** The number 2 falls into a cycle that does not include 1.
3. **Example 3:**
    - **Input:** `n = 1`
    - **Output:** `True`
    - **Explanation:** The number 1 is trivially a happy number as it already equals 1.
4. **Example 4:**
    - **Input:** `n = 20`

    - **Output:** `False`
    - **Explanation:** The number 20 falls into a cycle that does not include 1.

Explanation of Sample Input and Output:

- For the input `n = 19`, the function returns `True` because the number 19 eventually reaches 1 through the sum of the squares of its digits.
- For the input `n = 2`, the function returns `False` because the number 2 falls into a cycle that does not include 1.
- For the input `n = 1`, the function returns `True` because 1 is already 1, making it a happy number.
- For the input `n = 20`, the function returns `False` because 20 falls into a cycle that does not include 1.

Hints:

- Use a set to keep track of numbers you have already seen to detect cycles.
- Convert the number to its digits, square each digit, and sum the squares to get the next number in the sequence.