1. Write a program bn3that takes a list of integers and a target sum. Return all **unique pairs** of numbers from the list that add up to the target.

📥 Input Example:

```
nums = [2, 4, 3, 5, 6, -1]
target = 5
```

📤 Expected Output:

```
[(2, 3), (6, -1)]
```

Order doesn't matter. Pairs `(3, 2)` and `(2, 3)` are considered the same.

🧪 Test Cases:

```
Input: nums = [1, 2, 3, 4, 5], target = 6
Output: [(1, 5), (2, 4)]

Input: nums = [0, -1, 2, -3, 1], target = -2
Output: [(-3, 1)]

Input: nums = [3, 3, 3, 3], target = 6
Output: [(3, 3)]
```

2. A **Happy Number** is a number defined by the following process:

   1. Starting with any positive integer,
   2. Replace the number by the **sum of the squares of its digits**,
   3. Repeat the process until the number equals 1 (where it will stay), or it **loops endlessly** in a cycle that does **not** include 1.

If it ends in 1, it is a **Happy Number**

📥 Input Example:

```
19
```

📤 Expected Output:

```
True
```

Explanation:

- $1^2 + 9^2 = 82$
- $8^2 + 2^2 = 68$
- $6^2 + 8^2 = 100$
- $1^2 + 0^2 + 0^2 = 1$ ✅

🧪 Test Cases:

```
Input: 19 → Output: True
Input: 2 → Output: False
Input: 7 → Output: True
Input: 20 → Output: False
```