ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

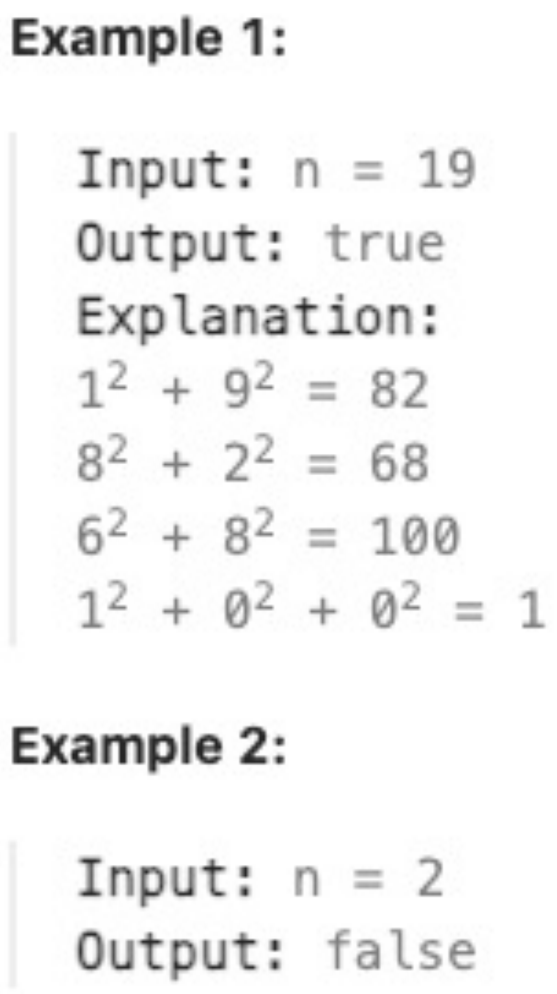
# ÔN TẬP LẬP TRÌNH

ONE LOVE. ONE FUTURE.

# Ex1. Happy Number

Write an algorithm to determine if a number n is happy.

- A **happy number** is a number defined by the following process:
  - Starting with any positive integer, replace the number by the sum of the squares of its digits.
  - Repeat the process until the number equals 1 (where it will stay), or it **loops endlessly in a cycle** which does not include 1.
  - Those numbers for which this process **ends in 1** are happy.
  - Return true *if* n *is a happy number, and* false *if not*.



**Example 1:**

Input: n = 19
Output: true
Explanation:
$1^2 + 9^2 = 82$
$8^2 + 2^2 = 68$
$6^2 + 8^2 = 100$
$1^2 + 0^2 + 0^2 = 1$

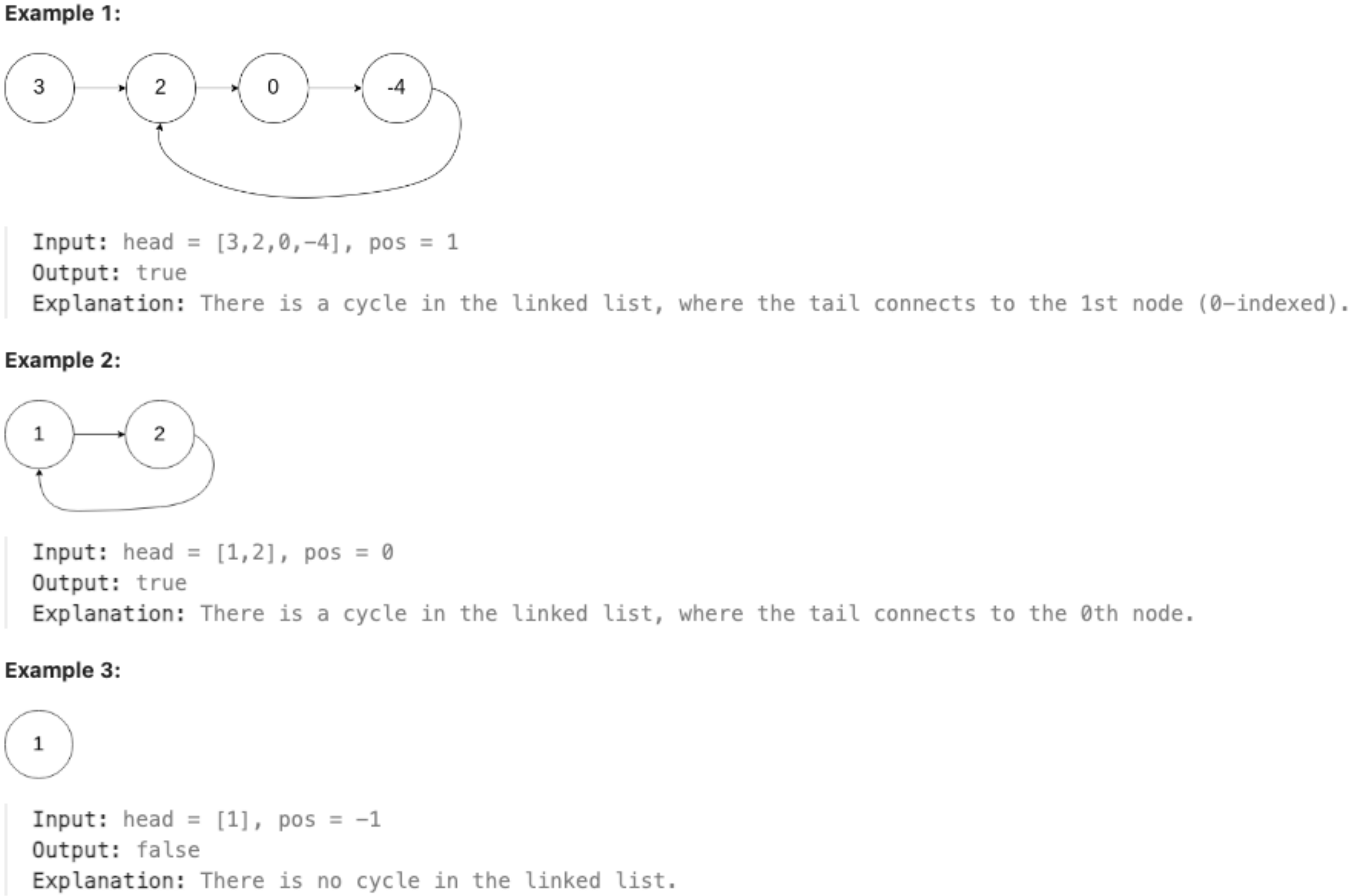
**Example 2:**

Input: n = 2
Output: false

# Ex2. Linked List Cycle

- Given head, the head of a linked list, determine if the linked list has a cycle in it.
- There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer. Internally, pos is used to denote the index of the node that tail's next pointer is connected to. **Note that pos is not passed as a parameter**.
- Return true *if there is a cycle in the linked list*. Otherwise, return false.

# Examples



**Example 1:**

```
Input: head = [3,2,0,-4], pos = 1
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 1st node (0-indexed).
```

**Example 2:**

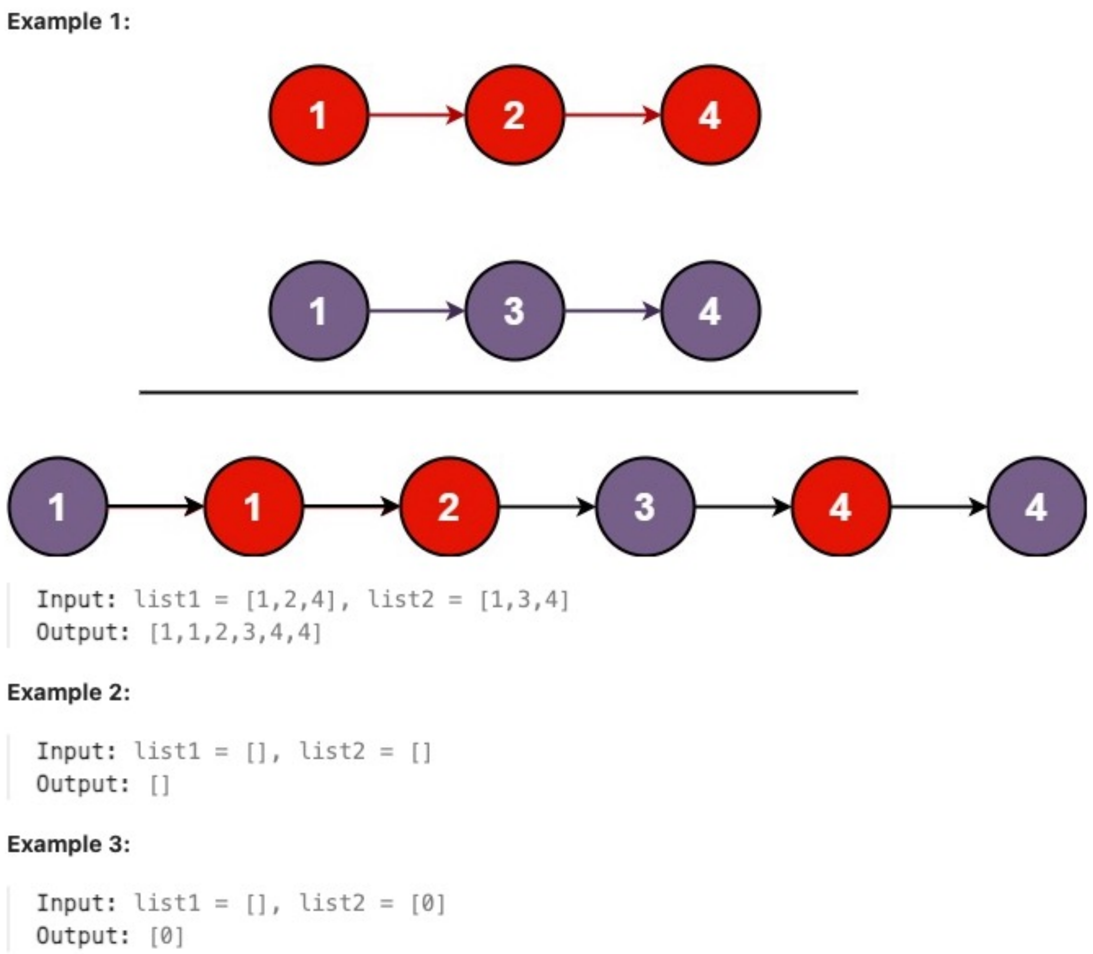
```
Input: head = [1,2], pos = 0
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 0th node.
```

**Example 3:**

```
Input: head = [1], pos = -1
Output: false
Explanation: There is no cycle in the linked list.
```

# Ex3. Merge Two Sorted Lists

- You are given the heads of two sorted linked lists list1 and list2.
- Merge the two lists into one **sorted** list. The list should be made by splicing together the nodes of the first two lists.
- Return *the head of the merged linked list*.



Example 1:

```
Input: list1 = [1,2,4], list2 = [1,3,4]
Output: [1,1,2,3,4,4]
```

Example 2:

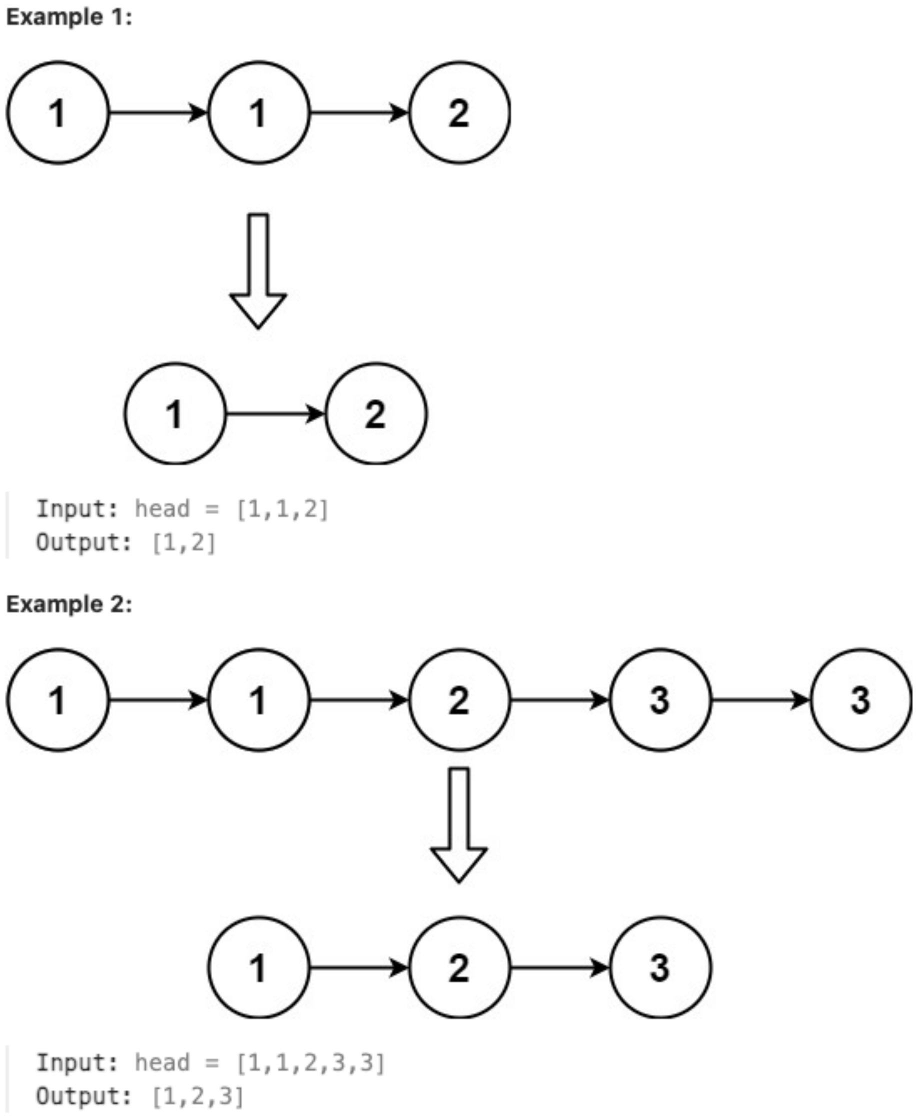
```
Input: list1 = [], list2 = []
Output: []
```

Example 3:

```
Input: list1 = [], list2 = [0]
Output: [0]
```

# Ex4. Remove Duplicates from Sorted List

- Given the head of a sorted linked list, *delete all duplicates such that each element appears only once*. Return *the linked list **sorted** as well*.



**Example 1:**

```
Input: head = [1,1,2]
Output: [1,2]
```

**Example 2:**

```
Input: head = [1,1,2,3,3]
Output: [1,2,3]
```

HUST

hust.edu.vn fb.com/dhbkhn

# THANK YOU !