CS204(2025)
Linked List - I

Instructions for Implementation

- Write the program in C or C++ using standard input/output.
- Follow the input/output format strictly.
- Ensure your code:
  - Handles edge cases correctly.
  - Meets the time and space complexity constraints.
- Do not use STL containers like vector, set, or map in C++. Use raw arrays unless explicitly allowed.
- You are encouraged to write clean, modular, and well-documented code.

Q1. Middle of the Linked List

Problem:

Given the head of a **singly linked list**, return the middle **node** of the linked list. If there are two middle nodes, return the second middle node.

Input:

- A sequence of integers representing the elements of a linked list terminated by #.

Output:

- Print the element at the middle of the linked list.

Constraints:

- 1 ≤ Number of nodes ≤ 10^5
- -10^9 ≤ Node Value ≤ 10^9

Example 1:

Input: 1 2 3 4 5 #

Output: 3

Example 2:

Input: 1 2 3 4 5 6 #

Output: 4

Explanation: Since the list has two middle nodes with values 3 and 4, we return the second one.

Q2. Longest Consecutive Run

Problem:

Given the **head** of a **singly linked list** of integers, find the number that has the **longest consecutive sequence of repetitions**.

If two or more numbers share the same maximum consecutive length, you should **return the largest of those numbers.**

Input:

- A sequence of integers representing the elements of a linked list terminated by #.

Output:

- Print the element that has the longest consecutive sequence of repetition.

Constraints:

- 1 ≤ Number of nodes ≤ 10^5
- -10^9 ≤ Node Value ≤ 10^9

Example 1:

Input: 1 1 4 4 2 2 2 3 #

Output: 2

Explanation:

The number 1 repeats consecutively 2 times.

The number 4 repeats consecutively 2 times.

The number 2 repeats consecutively 3 times.

The longest consecutive run is of length 3, which belongs to the number 2.

Example 2:

Input: 5 5 5 9 9 9 1 2 5 #

Output: 9

Explanation:

The number 5 has a consecutive run of length 3.

The number 9 also has a consecutive run of length 3.

There is a tie for the maximum length (3).

According to the rule, we must return the larger of the two numbers.

max (5, 9) = 9

Q3. Reverse a Linked List

Problem:

Given the head of a singly linked list, reverse the list, and return the reversed list.
Input:

- A sequence of integers representing the elements of a linked list terminated by #.

Output:

- A sequence of integers representing the elements of the linked list in reverse order

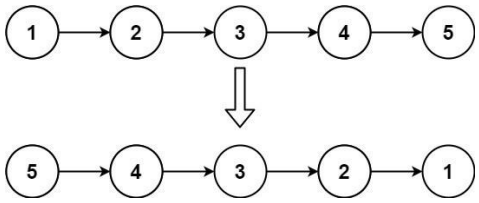
Constraints:

- 1 ≤ Number of nodes ≤ 10^5
- -10^9 ≤ Node Value ≤ 10^9

Example 1:

Input: 1 2 3 4 5 #

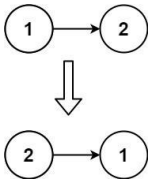
Output: 5 4 3 2 1 #



Example 2:

Input: 1 2 #

Output: 2 1 #



Q4. Maximum Twin Sum of a Linked List

Problem:

Given a linked list of size **n** (where n is **even)**

The i^{th} node (0-indexed) of the linked list is known as the twin of the $(n-i-1)^{th}$ node (0 <= i <= (n / 2) - 1)

For example, in **[A, B, C, D, E, F]** twins are **(A, F), (B, E), (C, D) ,**

The twin sum is defined as the sum of a node and its twin.

Given the head of a linked list with even length, return the maximum twin sum of the linked list.

Input:

- A sequence of integers representing the elements of a linked list terminated by #.

Output:

- Print the maximum twin sum of the element.

Constraints:

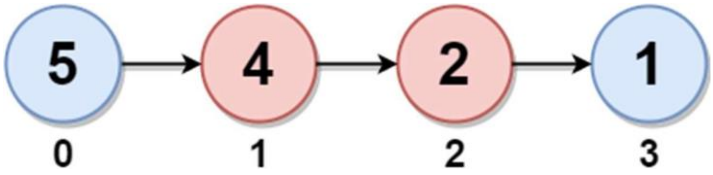
- 1 ≤ Number of nodes ≤ 10^5
- -10^9 ≤ Node Value ≤ 10^9

(Hint: Reverse the 2nd half of the LL)

Example 1:

Input: 5 4 2 1 #

Output: 6



Explanation:

Nodes 0 and 1 are the twins of nodes 3 and 2, respectively. All have twin sum = 6.

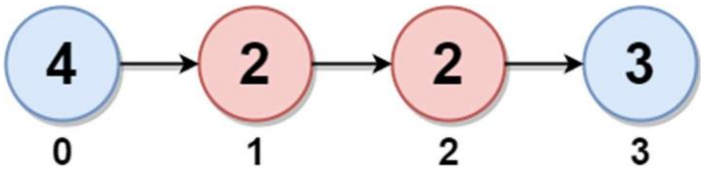
There are no other nodes with twins in the linked list.

Thus, the maximum twin sum of the linked list is 6.

Example 2:

Input: 4 2 2 3 #

Output: 7



Explanation:

The nodes with twins present in this linked list are:

- Node 0 is the twin of node 3 having a twin sum of 4 + 3 = 7.

Node 1 is the twin of node 2 having a twin sum of 2 + 2 = 4.

Thus, the maximum twin sum of the linked list is max (7, 4) = 7.

Q5. Rotate Linked List

Problem:

Given a linked list, **rotate** the list to the **right** by k places.
Input:

- The first line is a sequence of integers representing the elements of a linked list terminated by #.
- The next is the value of k.

Output:

- A sequence of integers representing the linked list, right rotated by k places.

Constraints:

- 1 ≤ Number of nodes ≤ 10^5
- -10^9 ≤ Node Value ≤ 10^9

Example 1:

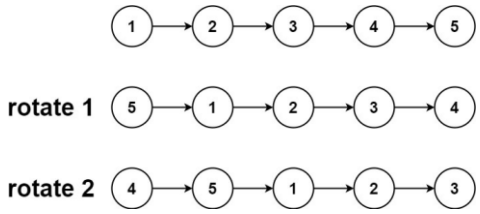
Input:

1 2 3 4 5 #

2

Output:

4 5 1 2 3



Example 2:

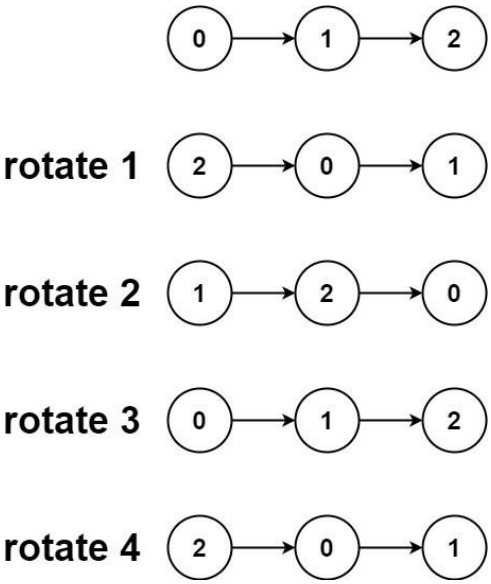
Input:

0 1 2 #

4

Output:

2 0 1



Q6. Find length of loop/cycle in given Linked List

Problem:

Given a linked list. The task is to find the length of the loop in the linked list. If the loop is not present, print 0.

Input:

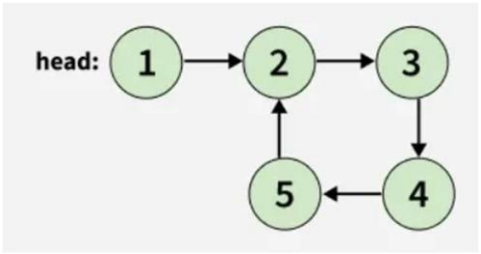
- First line is a sequence of integers representing the node value of a linked list terminated by #
- The subsequent lines consist of the node pair having connection (linked).

Output:

- the length of the cycle (int)

Constraints:

- 1 ≤ Number of nodes ≤ 10^5
- -10^9 ≤ Node Value ≤ 10^9



Example 1:

Input:

1 2 3 4 5 #

1 2

2 3

3 4

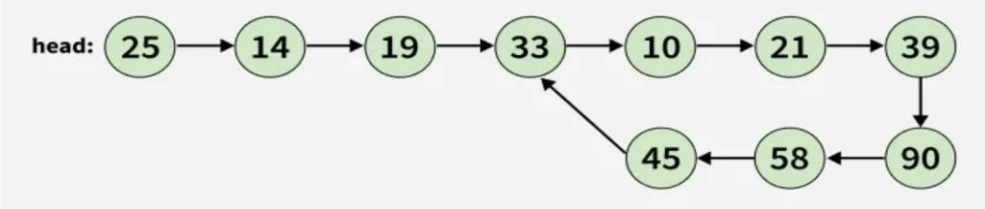
4 5

5 2

Output:

4

Explanation: There exists a loop in the linked list and the length of the loop is 4.



Example 2:

Input:

25 14 19 33 10 21 39 90 58 45 #

25 14

14 19

19 33

33 10

10 21

21 39

39 90

90 58

58 45

45 33

Output:

7

Explanation: The loop is present in the below-linked list, and the length of the loop is 7.

Q7. Odd Even Linked List

Given the head of a singly linked list, group all the nodes with odd indices together followed by the nodes with even indices and return the reordered list.

The first node is considered odd, and the second node is even, and so on.

Note that the relative order inside both the even and odd groups should remain as it was in the input.

You must solve the problem in O (1) extra space complexity and O(n) time complexity.

Input:

- A sequence of integers representing the elements of a linked list terminated by #.

Output:

- Linked list grouped by odd indices followed by even indices.

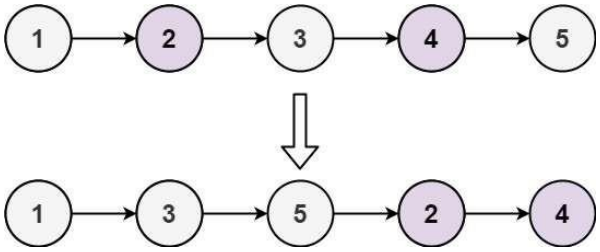
Constraints:

- 1 ≤ Number of nodes ≤ 10^5
- -10^9 ≤ Node Value ≤ 10^9

Example 1:

Input: 1 2 3 4 5 #

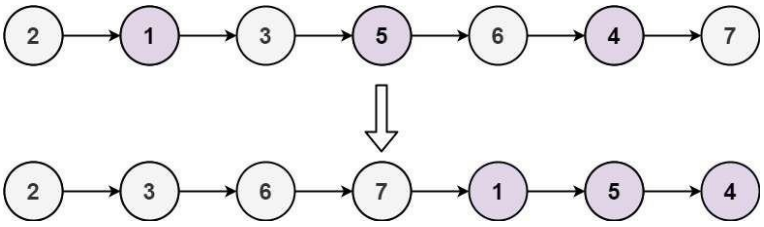
Output: 1 3 5 2 4



Example 2:

Input: 2 1 3 5 6 4 7 #

Output: 2 3 6 7 1 5 4



Q8. Rotate a LL

Problem:

Given a singly linked list and an integer k, the task is to rotate the linked list to the left by k places.

Input:

- The first line is a sequence of integers representing the elements of a linked list terminated by #.
- The next line is the value of the k.

Output:

- A sequence of integers representing the linked list, left rotated by k places.

Constraints:

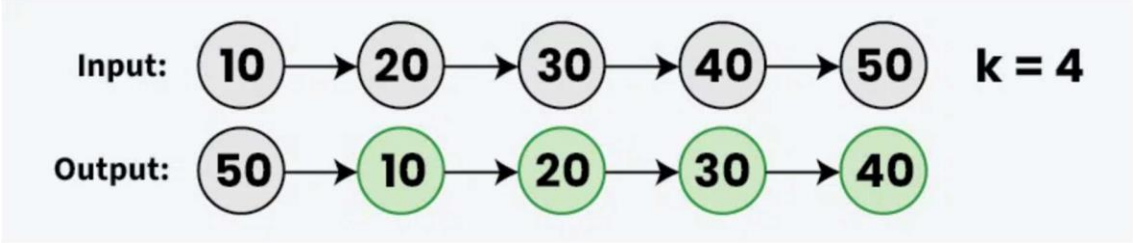
- 1 ≤ Number of nodes ≤ 10^5
- -10^9 ≤ Node Value ≤ 10^9

Example 1:

Input:
10 20 30 40 50 #

4

Output: 50 10 20 30 40



Explanation: After rotating the linked list to the left by 4 places, the 5th node, i.e node 50 becomes the head of the linked list and next pointer of node 50 points to node 10.

Input:

10 20 30 40 #

6

Output: 30 40 10 20