Q1: The Palindrome Check

Problem Description:

Write a program to check if a Singly Linked List is a Palindrome . A palindrome list reads the same backward as forward.

- **Challenge:** You must solve this **in-place** (using O(1) extra space) or by reversing the links. You cannot simply copy the list to a string or array.
- *Hint:* Find the middle, reverse the second half of the list, and compare.

Input Format:

- Line 1: N (Number of nodes).
- Line 2: N integers.

Output Format:

- Print True if it is a palindrome, otherwise False.

Sample Test Case 1:

Input:
5
1 2 3 2 1

Output:
True

Sample Test Case 2:

Input:
4
10 20 30 10

Output:
False

Q2: Segregate Even and Odd Values

Problem Description:

Given a Singly Linked List, write a function to modify it such that all Even numbers appear before all Odd numbers.

- **Constraint:** The **relative order** of the even numbers and the odd numbers must remain the same as in the original list.
- *Example:* If the input is 17 -> 8 -> 12, the output should be 8 -> 12 -> 17(8 came before 12 originally, so it stays before 12).

Input Format:

- Line 1: N (Number of nodes).
- Line 2: N integers.

Output Format:

- Print the modified list.

Sample Test Case 1:

Input:
7
17 15 8 12 10 5 4

Output:
8 12 10 4 17 15 5

Sample Test Case 2:

Input:
3
1 3 5

Output:
1 3 5

Q3: Rotate List

Problem Description:

Write a program to rotate a Singly Linked List to the Right by K places.

- **Logic:** The tail of the list moves to the front. This happens K times.
- **Constraint:** K can be larger than the size of the list N.

Input Format:

- Line 1: N (Number of nodes).
- Line 2: N integers.
- Line 3: K (Number of rotations).

Output Format:

- Print the rotated list.

Sample Test Case 1:

Input:
5
10 20 30 40 50
2

Output:
40 50 10 20 30

Sample Test Case 2:

Input:
3
1 2 3
4

Output:
3 1 2

Q4: Insert After Two (Sum & Skip)

Problem Description:

Write a program to modify a Singly Linked List based on the following rules:

1. Traverse the list from the head.
2. Take the first two nodes (a pair), calculate their **sum** , and insert a **new node** containing this sum immediately after the second node. Skip the newly added node.
3. Move to the next available pair of nodes and repeat.

Input Format:

- Line 1: N (Size of the list)
- Line 2: N space-separated integers.

Output Format:

- Print the modified Linked List.

Sample Test Case 1:

Input:
4
2 4 5 3

Output:
2 4 6 5 3 8

Explanation:

- Pair (2, 4): Sum is 6. Insert 6 after 4. List: 2->4->6->5->3
- Skip the newly added node. Next pair (5, 3): Sum is 8. Insert 8 after 3. List:2->4->6->5->3->8

Sample Test Case 2:

Input:
3
1 3 7

Output:
1 3 4 7 7

*** **Underlined elements are the newly added nodes** ***

Q5: Swap Nodes in Pairs

Problem Description:

Write a program to swap every two adjacent nodes in a Singly Linked List.

- Example: 1 -> 2 -> 3 -> 4 becomes 2 -> 1 -> 4 -> 3.
- **Strict Constraint:** You must swap the actual **Node pointers** (next), not the data values inside the nodes.
- If the list has an odd number of nodes, the last node remains as is.

Input Format:

- Line 1: N (Number of nodes).
- Line 2: N integers.

Output Format:

- Print the swapped list.

Sample Test Case 1:

Input:
4
10 20 30 40

Output:
20 10 40 30

Sample Test Case 2:

Input:
5
1 2 3 4 5

Output:
2 1 4 3 5

Q6: Increasing Subsequence from First Element

Problem Description:

Create a singly linked list containing **N** nodes initialized with the given values. Starting from the head of the list, perform the following operation iteratively:

- If the value of the next node is **less than or equal to** the value of the current node, remove the next node from the list.
- Otherwise, move the pointer to the next node.

After completing this process for the entire list, output the modified linked list.

Input Format:

- Line 1: N (Number of elements)
- Line 2: N integers

Sample Test case 1:

Input:
4
5 3 6 1

Output:
5 -> 6

Sample Test case 2:
Input:
5
1 3 3 5 4

Output:
1 -> 3 -> 5

Q7: Move all Occurrences to the End

Problem Description:

Given a singly linked list and an integer X, rearrange the list so that all nodes containing value X are moved to the end of the list.
The relative order of the other nodes must remain unchanged.

Input Format:

- Line 1: N (Number of elements)
- Line 2: N integers
- Line 3: The integer X

Sample Test Case 1:
Input:
7
1 2 3 2 4 2 5
2

Output:
1 3 4 5 2 2 2

Sample Test Case 2:
Input:
5
9 9 9 1 2
9

Output:
1 2 9 9 9

Q8: Find new Element

Problem Description:

An integer array A of size N is given. A new array B is constructed by reshuffling the elements of array A and adding two new random elements. Hence, the size of array B is N + 2.
Find the new added elements.

Input Format:

- Line 1: N (Size of A).
- Line 2: N integers (Elements of B)
- Line 3: N + 2 integers (Elements of B)

Sample Test case 1:

Input:

3
1 2 3
3 6 2 3 1

Output:
3 6

Sample Test case 2:
Input:
5
7 4 5 1 5
1 3 2 7 5 5 4

Output:
2 3

Q9: Same Parity Count

Problem Description:

An integer array of size N is given. Count the number of times an even number appearing on an even index, and an odd number appearing on an odd index, when sorted in increasing order. (0-based indexing is followed)

Input Format:

- Line 1: N (Size of array).
- Line 2: N integers (Elements of array)

Sample Test case 1:
Input:
3
2 5 4

Output:
1, 0

Sample Test case 2:
Input:
5
3 6 4 5 1

Output:
2, 2

Q10: Maximum Elements in a Range

Problem Description:

An integer array of size N is given. The array may contain duplicate elements.
Find the maximum number of elements that can be chosen from the array such that the difference between the maximum and the minimum value is less than equal to K.

Input Format:

- Line 1: N (Size of array)
- Line 2: K (Maximum difference)
- Line 3: N integers (Elements of array)

Sample Test case 1:
Input:
5
1
2 1 1 3 2

Output:
4

Explanation: The values at index 0, 1, 2 and 4 will be chosen, as (2 - 1) ≤ 1

Sample Test case 2:
Input:
7
3
6 2 5 4 5 1 2

Output:
5

Explanation: The values at index 1, 2, 3, 4 and 6 will be chosen, as (5 - 2) ≤ 3