**Problem-1:**
You are given an array of size N. you have to find out $k^{th}$ largest element from the array. No element will be repeated twice.
Hint: Sort the array and print kth element.

Sample input:
N=10, K=4
Array={3, 7, 6, 4, 9, 11, 15, 12, 19, 5 }

Sample output:
The $4^{th}$ largest element is : 11

**Problem-2:**
You are given an array contains consequent n numbers if the first number is 1 then the last number will be 1+(n-1) of the array. you are asked to find a number k which is in between the range from the starting number to the ending number. now you have to find out how many minimum steps needed to find out the index of the number. First number will be always 1 and the last number will be n.
Hint: Binary search

Sample input:
N=100
K=13
Array={1,2,3,4,5,6, …….98,99,100}

Sample output:
The minimum steps will be needed is : 6
Note: it is guaranteed that the number k will be always present in the array since we said the array's first element is 1 and last element is n and the k will be in between 1 and n .

**Problem-3:**
You are given all numbers between 1,2,…,n except one number. Your task is to find the missing number.
The first input line contains an integer n.
The second line contains n−1 numbers. Each number is distinct and between 1 and n(inclusive).

Sample input:
N=5
Array={2,3,1,5}

Sample output:
The missing number is : 4

**Problem-4:**
You are given a list of n integers, and your task is to calculate the number of *distinct* values in the list. The first input line has an integer n: the number of values.
The second line has n integers $x1,x2,\ldots,x_n$
Print one integers: the number of distinct values.

Sample Input:
N=5
Array={2, 3, 2, 2, 3}

Sample Output:
There are total 2 distinct element.

**Problem-5:**
You are given an array of n element now you are asked to find the largest sum of adjacent two element.

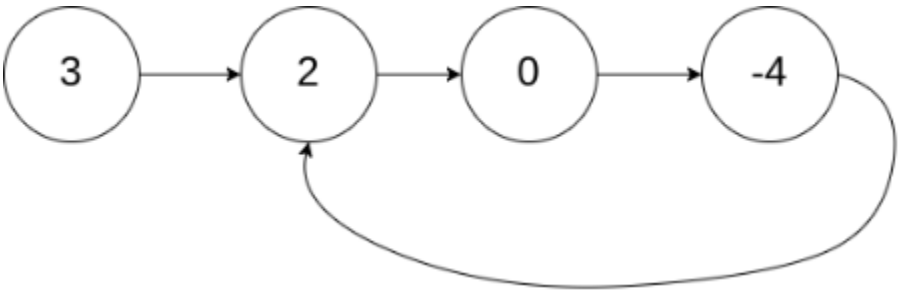
Sample input:
N=7
Array={1,6, 4, 8, 9, 2,12, 1 }

Sample output:
the maximum sum of two adjacent element is : 17

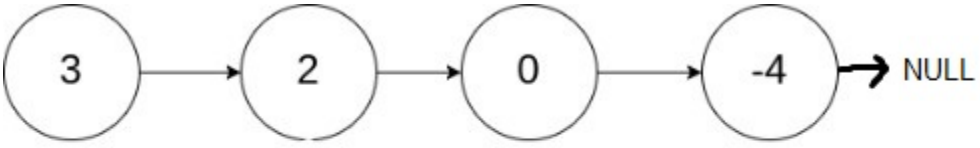
**Problem-6:**
You are given the head pointer and the size of a linked list determine if the linked list has a cycle in it. There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer.

hint: use the technique how you find out the size of a linked list check whether the while loop runs more than the size of an array or not.

Sample input:
N=4

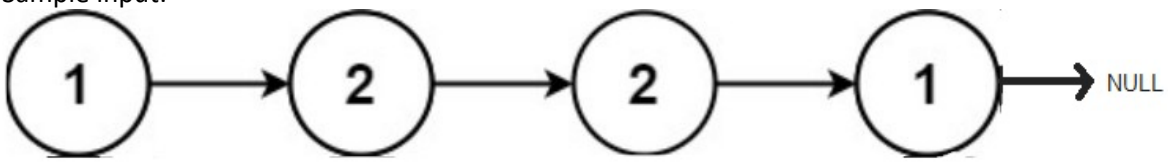



Sample output:
Yes the linked list has a loop.
Another sample input:
N=4

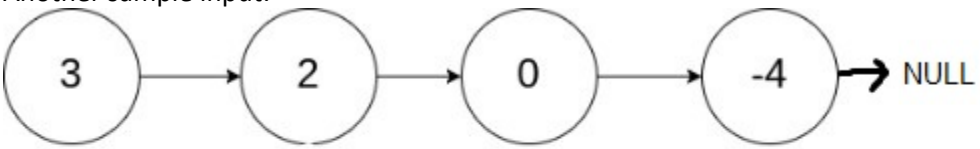



Sample output:
No, the linked list doesn't have a loop.

**Problem-7:**
You are given a head pointer of a singly linked list. You have to find out whether the linked list is palindrome or not.
Sample input:



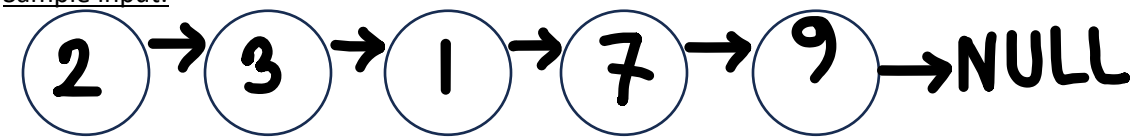

Sample output:
Yes the given linked list is palindrome.
Another sample input:




Sample output:
No, the linked list is not palindrome.

**Problem-8:**
You are given a linked list and the head pointer of the linked list. You are asked to find out the middle element of the linked list and print it. It is guaranteed that the length of the linked list is always odd.

Sample input:




Sample output:
The middle element of the linked list is : 1

**Problem-9:**
Given two sorted linked lists consisting of N and M nodes respectively. The task is to merge both of the lists (in place) and print the merged list.

Sample input:
N=3
M=4
A= 1->2->3->NULL
B= 4->5->6->7->NULL

Output:
C=1->2->3->4->5->6->7->NULL