**Algorithmic Thinking in Problem Solving**

**Problem 1: Middle of Linked List**

Given a non-empty, singly linked list with head node head, return a reference to the middle node of the linked list.

If the list has an even number of nodes, return the node at index n/2, where n is the number of nodes. For example, if n =4, return a references to the node at position 4/2 = 2

Write two solutions to this problem:

First solution: No constrains, just solve the problem

Second solution: Use one loop only

**Example 1:**

**Input:** 1 -> 2 -> 3 -> 4 -> 5

**Output:** Node 3 from this list

**Example 2:**

**Input:** 1 -> 2 -> 3 -> 4 -> 5 -> 6

**Output:** Node 4 from this list

**Problem 2: Convert Binary Number in a Linked List to Integer**

Given head which is a reference node to a singly-linked list. The value of each node in the linked list is either 0 or 1. The linked list holds the binary representation of a number.

Return the decimal value of the number in the linked list.

**Example 1:**

Diagram

Description automatically generated

**Input:** head = [1,0,1]

**Output:** 5

**Explanation:** (101) in base 2 = (5) in base 10

**Example 2:**

**Input:** head = [0]

**Output:** 0

**Example 3:**

**Input:** head = [1]

**Output:** 1

**Example 4:**

**Input:** head = [1,0,0,1,0,0,1,1,1,0,0,0,0,0,0]

**Output:** 18880

**Example 5:**

**Input:** head = [0,0]

**Output:** 0

**Constraints:**

* The Linked List is not empty.
* Number of nodes will not exceed 30.
* Each node's value is either 0 or 1

**Problem 3: Rotate Array**

Given an array, rotate the array to the right by *k* steps, where *k* is non-negative.

**Example 1:**

**Input:** [1,2,3,4,5,6,7] and *k* = 3

**Output:** [5,6,7,1,2,3,4]

**Explanation:**

rotate 1 steps to the right: [7,1,2,3,4,5,6]

rotate 2 steps to the right: [6,7,1,2,3,4,5]

rotate 3 steps to the right: [5,6,7,1,2,3,4]

**Example 2:**

**Input:** [-1,-100,3,99] and *k* = 2

**Output:** [3,99,-1,-100]

**Explanation:**

rotate 1 steps to the right: [99,-1,-100,3]

rotate 2 steps to the right: [3,99,-1,-100]

Write two solutions to this problem:

First solution: No constrains, just solve the problem

Second solution: Solve the problem in-place – O(1) extra space

**Problem 4: DI String Match**

Given a string S that **only** contains "I" (increase) or "D" (decrease), let N = S.length.

Return **any** permutation A of [0, 1, ..., N] such that for all i = 0, ..., N-1:

* If S[i] == "I", then A[i] < A[i+1]
* If S[i] == "D", then A[i] > A[i+1]

**Example 1:**

**Input:** "IDID"

**Output:** [0,4,1,3,2]

**Example 2:**

**Input:** "III"

**Output:** [0,1,2,3]

**Example 3:**

**Input:** "DDI"

**Output:** [3,2,0,1]

**Note:**

1. 1 <= S.length <= 10000
2. S only contains characters "I" or "D".

Spend at most 1 hour trying to solve each of the problems. If you are unable so solve the problem after 1 hour, Google the problem and find a solution, then do the following:

1. Trace the solution using a concrete instance of the problem
2. What prevented you from solving the problem?
3. What did you learn? Did you have to Google a little more to understand the solution?
4. What would you do differently in the future if you were presented with a similar problem?