

**NORTHERN UNIVERSITY OF BUSINESS AND TECHNOLOGY KHULNA**



**Lab Assignment**

**Assignment No: 02**

**Course Title:** Artificial Intelligence Lab

**Course Code:** CSE 4112

**SUBMITTED BY**

**Name:** Md. Mahfujul Karim Sheikh  
**ID:** 11220120735  
**Section:** 7B  
**Department:** CSE  
**Session:** Spring 2025

**SUBMITTED TO**

**Name:** Syed Md. Galib  
**Designation:** Professor  
**Department:** CSE  
**Institution:** JUST

---

**Date Of Submission: 28 April 2025**

## 1.Palindrome Number(9).

```
class Solution:

    def isPalindrome(self, x: int) -> bool:

        if x < 0:

            return False

        rev = 0

        y = x

        while y:

            rev = rev * 10 + y % 10

            y //= 10

        return rev == x
```

**Input:** x = 121

**Output:** true

## 2.Remove Element(27).

```
class Solution:

    def removeElement(self, nums: list[int], val: int) -> int:

        i = 0

        for num in nums:

            if num != val:

                nums[i] = num

                i += 1
```

```
return i
```

**Input:** nums = [0,1,2,2,3,0,4,2], val = 2

**Output: 5, nums = [0,1,4,0,3,\_,\_,\_]**

### 3.Length of Last Word(57)

```
class Solution:
```

```
def lengthOfLastWord(self, s: str) -> int:
```

```
    i = len(s) - 1
```

```
    while i >= 0 and s[i] == ' ':
```

```
        i -= 1
```

```
    lastIndex = i
```

```
    while i >= 0 and s[i] != ' ':
```

```
        i -= 1
```

```
    return lastIndex - i
```

**Input:** s = "Hello World"

**Output: 5**

### 4.Pascal's Triangle(118).

```
class Solution:
```

```
def generate(self, numRows: int) -> list[list[int]]:
```

```
    ans = []
```

```

for i in range(numRows):

    ans.append([1] * (i + 1))


for i in range(2, numRows):

    for j in range(1, len(ans[i]) - 1):

        ans[i][j] = ans[i - 1][j - 1] + ans[i - 1][j]


return ans

```

**Input:** numRows = 5

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

### 5.Valid Anagram(242)

```

class Solution:

    def isAnagram(self, s: str, t: str) -> bool:

        if len(s) != len(t):

            return False


        count = collections.Counter(s)

        count.subtract(collections.Counter(t))

        return all(freq == 0 for freq in count.values())

```

**Input:** s = "anagram", t = "nagaram"

**Output:** true

## 6.Find Peak Element (162).

```
class Solution:

    def findPeakElement(self, nums: list[int]) -> int:

        l = 0

        r = len(nums) - 1

        while l < r:

            m = (l + r) // 2

            if nums[m] >= nums[m + 1]:

                r = m

            else:

                l = m + 1

        return l
```

**INPUT:** nums = [1,2,1,3,5,6,4]

**OUTPUT:** 5

## 7.Binary Tree Inorder Traversal(94).

```
class Solution:

    def inorderTraversal(self, root: TreeNode | None) -> list[int]:

        ans = []

        stack = []

        while root or stack:

            while root:

                stack.append(root)

                root = root.left
```

```

    root = root.left

    root = stack.pop()

    ans.append(root.val)

    root = root.right

    return ans

```

**Input:** root = [1,null, 2,3]

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

## 8. Word Search(79).

```

class Solution:

    def exist(self, board: list[list[str]], word: str) -> bool:

        m = len(board)

        n = len(board[0])

        def dfs(i: int, j: int, s: int) -> bool:

            if i < 0 or i == m or j < 0 or j == n:

                return False

            if board[i][j] != word[s] or board[i][j] == '*':

                return False

            if s == len(word) - 1:

                return True

            cache = board[i][j]

            board[i][j] = '*'

```

```
isExist = (dfs(i + 1, j, s + 1) or
```

```
    dfs(i - 1, j, s + 1) or
```

```
    dfs(i, j + 1, s + 1) or
```

```
    dfs(i, j - 1, s + 1))
```

```
board[i][j] = cache
```

```
return isExist
```

```
return any(dfs(i, j, 0)
```

```
    for i in range(m)
```

```
    for j in range(n))
```

**Input:** board = [["A","B","C","E"],["S","F","C","S"],["A","D","E","E"]], word = "ABCCED"

**Output:** true

## 9. Binary Tree Level Order Traversal(102).

```
class Solution:
```

```
    def levelOrder(self, root: TreeNode | None) -> list[list[int]]:
```

```
        if not root:
```

```
            return []
```

```
        ans = []
```

```
        q = collections.deque([root])
```

```

while q:

    currLevel = []

    for _ in range(len(q)):

        node = q.popleft()

        currLevel.append(node.val)

        if node.left:

            q.append(node.left)

        if node.right:

            q.append(node.right)

    ans.append(currLevel)

return ans

```

**Input:** root = [3,9,20,null,null,15,7]

**Output:** [[3],[9,20],[15,7]]

## 10. Path Sum(112).

```

class Solution:

    def hasPathSum(self, root: TreeNode, summ: int) -> bool:

        if not root:

            return False

        if root.val == summ and not root.left and not root.right:

            return True

        return (self.hasPathSum(root.left, summ - root.val) or


```






```
self.hasPathSum(root.right, summ - root.val))
```

**Input:** root = [5,4,8,11,null,13,4,7,2,null,null,null,1], targetSum = 22

**Output:** true

 [Explore](#) [Problems](#) [Contest](#) [Discuss](#) [Interview](#) [Store](#)

  0  [Premium](#)

All My Submissions

To exit full screen, press and hold Esc

Time Submitted	Question	Status	Runtime	Language
0 minutes ago	<a href="#">Path Sum</a>	Accepted	2 ms	python3
27 minutes ago	<a href="#">Binary Tree Level Order Traversal</a>	Accepted	0 ms	python3
43 minutes ago	<a href="#">Word Search</a>	Accepted	3133 ms	python3
13 hours, 15 minutes ago	<a href="#">Binary Tree Inorder Traversal</a>	Accepted	0 ms	python3
13 hours, 55 minutes ago	<a href="#">Find Peak Element</a>	Accepted	0 ms	python3
14 hours, 3 minutes ago	<a href="#">Valid Anagram</a>	Accepted	8 ms	python3
1 day, 1 hour ago	<a href="#">Pascal's Triangle</a>	Accepted	0 ms	python3
1 day, 1 hour ago	<a href="#">Length of Last Word</a>	Accepted	0 ms	python3
1 day, 2 hours ago	<a href="#">Palindrome Number</a>	Accepted	11 ms	python3
1 day, 2 hours ago	<a href="#">Remove Element</a>	Accepted	0 ms	python3

Newer

Older