**DAA Assignment 2 Huffman Encoding**

**Code:**

class Node:

    def \_\_init\_\_(self, char, freq):

        self.char = char

        self.freq = freq

        self.left = None

        self.right = None

        self.huff = ''

def printNodes(node, val=''):

    newVal = val + str(node.huff)

    if node.left:

        printNodes(node.left, newVal)

    if node.right:

        printNodes(node.right, newVal)

    if not node.left and not node.right:

        print(f"{node.char} -> {newVal}")

def main():

    chars = []

    freq = []

    num\_chars = int(input("Enter the number of characters: "))

    for \_ in range(num\_chars):

        char = input("Enter character: ")

        frequency = int(input(f"Enter frequency for {char}: "))

        chars.append(char)

        freq.append(frequency)

    nodes = []

    for x in range(len(chars)):

        nodes.append(Node(chars[x], freq[x]))

    while len(nodes) > 1:

        nodes = sorted(nodes, key=lambda x: x.freq)

        left = nodes[0]

        right = nodes[1]

        left.huff = 0

        right.huff = 1

        newNode = Node(left.char + right.char, left.freq + right.freq)

        newNode.left = left

        newNode.right = right

        nodes.remove(left)

        nodes.remove(right)

        nodes.append(newNode)

    printNodes(nodes[0])

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

Enter the number of characters: 4

Enter character: a

Enter frequency for a: 4

Enter character: b

Enter frequency for b: 7

Enter character: c

Enter frequency for c: 12

Enter character: d

Enter frequency for d: 14

d -> 0

a -> 100

b -> 101

c -> 11