**Code -**

import hashlib

class Node:

def \_\_init\_\_(self, hash\_value, value=None, left=None, right=None):

self.hash\_value = hash\_value

self.value = value

self.left = left

self.right = right

def create\_merkle\_tree(hashes, values):

if len(hashes) == 1:

return Node(hashes[0], values[0])

else:

return Node(hashlib.sha256("".join(hashes).encode()).hexdigest(), values, create\_merkle\_tree(hashes[:len(hashes)//2], values[:len(values)//2]), create\_merkle\_tree(hashes[len(hashes)//2:], values[len(values)//2:]))

def print\_merkle\_tree(tree):

if tree.left:

print\_merkle\_tree(tree.left)

if tree.right:

print\_merkle\_tree(tree.right)

print("Hash value:",tree.hash\_value, "Transactions:", tree.value)

number\_of\_transactions = int(input("Enter the number of transactions: "))

transactions = []

hashes = []

for i in range(number\_of\_transactions):

transaction = input("Enter the transaction: ")

transactions.append(transaction)

hashes.append(hashlib.sha256(transaction.encode()).hexdigest())

if len(transactions) % 2 != 0:

transactions.append(transactions[-1])

hashes.append(hashes[-1])

merkle\_tree = create\_merkle\_tree(hashes, transactions)

print\_merkle\_tree(merkle\_tree)

**Output -**

