

Code:

```
1 from typing import List
2 import hashlib
3
4
5 class Node:
6     def __init__(self, left, right, value: str, content, is_copied=False) ->
        None:
7         self.left: Node = left
8         self.right: Node = right
9         self.value = value
10        self.content = content
11        self.is_copied = is_copied
12
13    @staticmethod
14    def hash(val: str) -> str:
15        return hashlib.sha256(val.encode('utf-8')).hexdigest()
16
17    def __str__(self):
18        return (str(self.value))
19
20    def copy(self):
21        """
22        class copy function
23        """
24        return Node(self.left, self.right, self.value, self.content, True)
25
```

```

26
27 class MerkleTree:
28     def __init__(self, values: List[str]) -> None:
29         self.__buildTree(values)
30
31     def __buildTree(self, values: List[str]) -> None:
32
33         leaves: List[Node] = [Node(None, None, Node.hash(e), e)
34                                for e in values]
35         if len(leaves) % 2 == 1:
36
37             leaves.append(leaves[-1].copy())
38         self.root: Node = self.__buildTreeRec(leaves)
39
40     def __buildTreeRec(self, nodes: List[Node]) -> Node:
41         if len(nodes) % 2 == 1:
42             # duplicate last elem if odd number of elements
43             nodes.append(nodes[-1].copy())
44             half: int = len(nodes) // 2
45
46         if len(nodes) == 2:
47             return Node(nodes[0], nodes[1], Node.hash(nodes[0].value + nodes[1]
48                                                         .value), nodes[0].content+"-"+nodes[1].content)
49
50         left: Node = self.__buildTreeRec(nodes[:half])
51         right: Node = self.__buildTreeRec(nodes[half:])

```

```

51         value: str = Node.hash(left.value + right.value)
52         content: str = f'{left.content}+{right.content}'
53         return Node(left, right, value, content)
54
55     def printTree(self) -> None:
56         self.__printTreeRec(self.root)
57
58     def __printTreeRec(self, node: Node) -> None:
59         if node != None:
60             if node.left != None:
61                 print("Left: "+str(node.left))
62                 print("Right: "+str(node.right))
63             else:
64                 print("Input")
65
66             if node.is_copied:
67                 print('(Padding)')
68                 print("Value: "+str(node.value))
69                 print("Content: "+str(node.content))
70                 print("")
71                 self.__printTreeRec(node.left)
72                 self.__printTreeRec(node.right)
73
74     def getRootHash(self) -> str:
75         return self.root.value
76

```

```

76
77
78 def mixmerkletree() -> None:
79     elems = ["Aztec", "A", "Team",
80             "Om", "Hamza", "Mohib", "Parth"]
81
82     print("Inputs: ")
83     print(*elems, sep=" | ")
84     print("")
85     mtree = MerkleTree(elems)
86     print("Root Hash: "+mtree.getRootHash()+"\n")
87     mtree.printTree()
88
89
90 mixmerkletree()

```

Output:

Inputs:

Aztec | A | Team | Om | Hamza | Mohib | Parth

Root Hash: 701fa83df0163756a4c8bd11ab5e69f5a763a386e29cb79265ec29ef840f4685

Left: 5a806abd172de56e2d43ebe13aece485e54f600361901219d788d0750913837b

Right: 545cf9a5a92db7affd5a4a20a4168337e485d70a195e13e035dc793923c8c7ab

Value: 701fa83df0163756a4c8bd11ab5e69f5a763a386e29cb79265ec29ef840f4685

Content: Aztec+A+Team+Om+Hamza+Mohib+Parth+Parth

Left: 7ef6c3416f199dde05fa2eec22864ecf829b7fb1f7fc97075e9e4958febc40ad

Right: c527ffbf4fe41d3285a20237c97b2a0f02accfecbd66dc42555f62f10d3a95c1

Value: 5a806abd172de56e2d43ebe13aece485e54f600361901219d788d0750913837b

Content: Aztec+A+Team+Om

Left: d28a9d9c11188360cc63bceaef700835526a239329056daffa0cf1a6a856d4c

Right: 559aead08264d5795d3909718cdd05abd49572e84fe55590eef31a88a08fdffd

Value: 7ef6c3416f199dde05fa2eec22864ecf829b7fb1f7fc97075e9e4958febc40ad

Content: Aztec+A

Input

Value: d28a9d9c11188360cc63bceaef700835526a239329056daffa0cf1a6a856d4c

Content: Aztec

Input

Value: 559aead08264d5795d3909718cdd05abd49572e84fe55590eef31a88a08fdffd

Input

Value: 559aead08264d5795d3909718cdd05abd49572e84fe55590eef31a88a08fdffd

Content: A

Left: 5985039f106df054b43c3529139613bf7c2372937539f344c7c5ee47d00a0d63

Right: 3c815b4770735e66231e08cb8865b50308ba7301c579d38c171f4ebec5e3e9fc

Value: c527ffbf4fe41d3285a20237c97b2a0f02accfecbd66dc42555f62f10d3a95c1

Content: Team+Om

Input

Value: 5985039f106df054b43c3529139613bf7c2372937539f344c7c5ee47d00a0d63

Content: Team

Input

Value: 3c815b4770735e66231e08cb8865b50308ba7301c579d38c171f4ebec5e3e9fc

Content: Om

Left: dcd78e1ce4bea5aa05d146cc658bafff7bdf75d7f19042ea90b9b4382618f0

Right: 2c3b263e732b8c8a97c01b960d3e9e84e0fb7283a035f87d7dc1152d3468d8b0

Value: 545cf9a5a92db7affd5a4a20a4168337e485d70a195e13e035dc793923c8c7ab

Content: Hamza+Mohib+Parth+Parth

Left: 26ac05af275df37b36a23a75dfb701c48ecbdb6a5a578674c36274737864f4ce

Right: f3c95de41483179952d2b4aef3811383ac2deae8c33e708e764d81856d29f5c3

Value: dcd78e1ce4bea5aa05d146cc658bafff7bdf75d7f19042ea90b9b4382618f0

Left: dcd78e1ce4bea5aa05d146cc658bafff7bdfff75d7f19042ea90b9b4382618f0
Right: 2c3b263e732b8c8a97c01b960d3e9e84e0fb7283a035f87d7dc1152d3468d8b0
Value: 545cf9a5a92db7affd5a4a20a4168337e485d70a195e13e035dc793923c8c7ab
Content: Hamza+Mohib+Parth+Parth

Left: 26ac05af275df37b36a23a75dfb701c48ecbdb6a5a578674c36274737864f4ce
Right: f3c95de41483179952d2b4aef3811383ac2deae8c33e708e764d81856d29f5c3
Value: dcd78e1ce4bea5aa05d146cc658bafff7bdfff75d7f19042ea90b9b4382618f0
Content: Hamza+Mohib

Input

Value: 26ac05af275df37b36a23a75dfb701c48ecbdb6a5a578674c36274737864f4ce
Content: Hamza

Input

Value: f3c95de41483179952d2b4aef3811383ac2deae8c33e708e764d81856d29f5c3
Content: Mohib

Left: 4ec1a76282e4f89a809b90cea83ef509a5eda8d8d128819905a7459830a2ebe5
Right: 4ec1a76282e4f89a809b90cea83ef509a5eda8d8d128819905a7459830a2ebe5
Value: 2c3b263e732b8c8a97c01b960d3e9e84e0fb7283a035f87d7dc1152d3468d8b0

Input

Value: f3c95de41483179952d2b4aef3811383ac2deae8c33e708e764d81856d29f5c3

Content: Mohib

Left: 4ec1a76282e4f89a809b90cea83ef509a5eda8d8d128819905a7459830a2ebe5

Right: 4ec1a76282e4f89a809b90cea83ef509a5eda8d8d128819905a7459830a2ebe5

Value: 2c3b263e732b8c8a97c01b960d3e9e84e0fb7283a035f87d7dc1152d3468d8b0

Content: Parth+Parth

Input

Value: 4ec1a76282e4f89a809b90cea83ef509a5eda8d8d128819905a7459830a2ebe5

Content: Parth

Input

(Padding)

Value: 4ec1a76282e4f89a809b90cea83ef509a5eda8d8d128819905a7459830a2ebe5

Content: Parth

=== Code Execution Successful ===