ZOHO

NAME: MALLESHA M

COLLAGE: GOVERNMENT ENGINEARING COLLAGE HASSAN

ROLE HIERACHY

```
from collections import deque


class Node:
    def _init_(self, data):
        self.data = data
        self.childrens = []



def printTree(node):
    q = deque()

    q.append(node)

    while len(q):
        item = q.popleft()
        print(item.data)
        for child in item.childrens:
            q.append(child)



def updateTree(rootNode, oldNode, newNode):
    if rootNode.data == oldNode.data:
```

```
        rootNode.data = newNode.data
        print("touch" + newNode.data)
        return rootNode.childrens
    else:
        for child in rootNode.childrens:
            print(child.data)
            updateTree(child, oldNode, newNode)



# Driver code
if _name_ == '_main_':

    myDict = {}
    rootRoleName = input("Enter root role name : ")
    #root node
    myDict[rootRoleName] = Node(rootRoleName)

    while True :
        print("Operations:")
        print("1. Add Sub Role.")
        print("2. Display Roles.")
        print("3. Delete Role.")
        # print("4. Add User.")
        # print("5. Display Users.")
        # print("6. Display Users and Sub Users.")
        # print("7. Delete User.")
        # print("8. Number of users from top.")
        # print("9. Height of role hierachy.")
        # print("10. Common boss of users.")
```

```
choice = int(input("Operation to be performed : "))


if choice == 1:
    subRoleName = input("Enter sub role name : ")
    reportingToRoleName = input("Enter reporting to role name : ")
    #new node creation
    myDict[subRoleName] = Node(subRoleName)
    # connecting new node with parent node
    myDict[reportingToRoleName].childrens.append(myDict[subRoleName])


elif choice == 2:
    printTree(myDict[rootRoleName])


elif choice == 3:
    roleToBeDeleted = input("Enter the role to be delelted : ")
    roleToBeTransferred = input("Enter the role to be transferred : ")


    # new node creation to delete and transfer old node
    myDict[roleToBeTransferred] = Node(roleToBeTransferred)


    updateTree(myDict[rootRoleName], myDict[roleToBeDeleted], myDict[roleToBeTransferred])


else:
    break
```