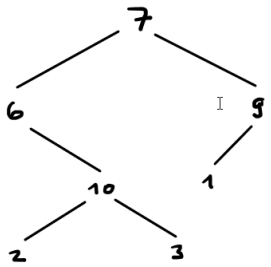
1. (5 Punkte) a. Schreibe eine Klasse `Node`, die einen Knoten eines binären Baums repräsentiert.

   b. Schreibe eine Funktion `baum`, die den abgebildeten Baum zurückgibt. Nutze dazu deine Node-Klasse. Verschachtele die Node-Aufrufe aus Lesbarkeitsgründen höchstens einmal und gib den temporären Teilbäumen sprechende Namen.



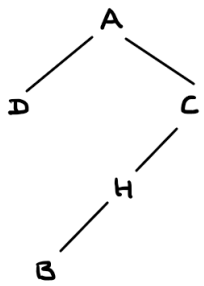
Lösung:

```
class Node:
    def __init__(self, val, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def baum():
    b10 = Node(10,Node(2),Node(3))
    b6 = Node(6,None,b10)
    b9 = Node(9,Node(1),None)
    b7 = Node(7,b6,b9)
    return b7
```

2. (5 Punkte) a. Schreibe eine Klasse `Node`, die einen Knoten eines binären Baums repräsentiert.

   b. Schreibe eine Funktion `baum`, die den abgebildeten Baum zurückgibt. Nutze dazu deine Node-Klasse. Verschachtele die Node-Aufrufe aus Lesbarkeitsgründen höchstens einmal und gib den temporären Teilbäumen sprechende Namen.



Lösung:

```
class Node:
    def __init__(self, val, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def baum():
    h = Node('H',Node('B'),None)
    c = Node('C',h,None)
    a = Node('A',Node('D'),c)
    return a
```