Source Code :

```
import heapq
class node:
    def __init__(self, freq, symbol, left=None, right=None):

        self.freq = freq
        self.symbol = symbol
        self.left = left
        self.right = right
        self.huff = ''

    def __lt__(self, nxt):
        return self.freq < nxt.freq

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.symbol} -> {newVal}")

chars = []
freq = []
nodes = []

m=int(input("enter the no of characters :"))
for i in range(m):
    s=input("enter the character"+str(i+1)+":")
    k=int(input("enter the freq for character"+str(i+1)+":"))
```

```
    chars.append(s)
    freq.append(k)

for x in range(len(chars)):
    heapq.heappush(nodes, node(freq[x], chars[x]))

while len(nodes) > 1:
    left = heapq.heappop(nodes)
    right = heapq.heappop(nodes)

    left.huff = 0
    right.huff = 1

    newNode = node(left.freq+right.freq, left.symbol+right.symbol, left,
right)
    heapq.heappush(nodes, newNode)

printNodes(nodes[0])
```

OUTPUT :