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

    def \_\_lt\_\_(self, other):

        return self.freq < other.freq

def generate\_huffman\_tree(chars, freq):

    pq = []

    # Create priority queue of nodes

    for i in range(len(chars)):

        heapq.heappush(pq, Node(freq[i], chars[i]))

    while len(pq) > 1:

        # Remove two nodes with lowest frequency

        left = heapq.heappop(pq)

        right = heapq.heappop(pq)

        # Create a new internal node with combined frequency

        new\_node = Node(left.freq + right.freq, None, left, right)

        # Push the new node back into the priority queue

        heapq.heappush(pq, new\_node)

    # Return the root of the Huffman tree

    return pq[0]

def generate\_huffman\_codes(node, current\_code, huffman\_codes):

    if node.symbol is not None:

        huffman\_codes[node.symbol] = current\_code

    else:

        generate\_huffman\_codes(node.left, current\_code + "0", huffman\_codes)

        generate\_huffman\_codes(node.right, current\_code + "1", huffman\_codes)

def print\_huffman\_codes(root):

    huffman\_codes = {}

    generate\_huffman\_codes(root, "", huffman\_codes)

    print("Huffman Codes:")

    for symbol, code in huffman\_codes.items():

        print(f"{symbol} -> {code}")

def main():

    n = int(input("Enter number of characters: "))

    chars = []

    freq = []

    for i in range(n):

        char = input(f"Enter character {i + 1}: ")

        chars.append(char)

        f = int(input(f"Enter frequency for {char}: "))

        freq.append(f)

    root = generate\_huffman\_tree(chars, freq)

    print\_huffman\_codes(root)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Output:

Enter number of characters: 4

Enter character 1: a

Enter frequency for a: 5

Enter character 2: b

Enter frequency for b: 9

Enter character 3: c

Enter frequency for c: 12

Enter character 4: d

Enter frequency for d: 13

Huffman Codes: a -> 00 b -> 01 c -> 10 d -> 11