

Code :

```
import java.util.PriorityQueue;  
import java.util.Scanner;  
  
// Node class representing each character and frequency  
class Node implements Comparable<Node> {  
    char ch;  
    int freq;  
    Node left, right;  
  
    Node(char ch, int freq) {  
        this.ch = ch;  
        this.freq = freq;  
    }  
  
    // Compare nodes by frequency (for priority queue)  
    public int compareTo(Node other) {  
        return this.freq - other.freq;  
    }  
}  
  
public class HuffmanEncoding {  
  
    // Recursive function to print Huffman codes from the root  
    static void printCodes(Node root, String code) {  
        if (root == null)  
            return;  
    }
```

```

// If leaf node, print character and its code
if (root.left == null && root.right == null && Character.isLetter(root.ch)) {
    System.out.println(root.ch + ":" + code);
    return;
}

// Traverse left and right
printCodes(root.left, code + "0");
printCodes(root.right, code + "1");
}

// Huffman encoding using Greedy method
static void huffmanEncode(char[] chars, int[] freq) {
    // Step 1: Create a priority queue (min-heap)
    PriorityQueue<Node> pq = new PriorityQueue<>();

    // Step 2: Create leaf nodes for each character
    for (int i = 0; i < chars.length; i++) {
        pq.add(new Node(chars[i], freq[i]));
    }

    // Step 3: Build Huffman tree
    while (pq.size() > 1) {
        Node left = pq.poll(); // smallest freq
        Node right = pq.poll(); // next smallest freq

        Node newNode = new Node('-', left.freq + right.freq);
        newNode.left = left;
        newNode.right = right;
    }
}

```

```
    pq.add(newNode); // push the combined node back to queue
}

// Step 4: The remaining node is the root of Huffman tree
Node root = pq.peek();

System.out.println("\nHuffman Codes:");
printCodes(root, "");

}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter number of characters: ");
    int n = sc.nextInt();

    char[] chars = new char[n];
    int[] freq = new int[n];

    System.out.println("Enter characters and their frequencies:");
    for (int i = 0; i < n; i++) {
        System.out.print("Character " + (i + 1) + ": ");
        chars[i] = sc.next().charAt(0);
        System.out.print("Frequency of " + chars[i] + ": ");
        freq[i] = sc.nextInt();
    }

    huffmanEncode(chars, freq);
    sc.close();
}
```

```
    }  
}
```

Output :

Enter number of characters: 6

Enter characters and their frequencies:

Character 1: a

Frequency of a: 5

Character 2: b

Frequency of b: 9

Character 3: c

Frequency of c: 12

Character 4: d

Frequency of d: 13

Character 5: e

Frequency of e: 16

Character 6: f

Frequency of f: 45

Huffman Codes:

f : 0

c : 100

d : 101

a : 1100

b : 1101

e : 111