package lp3;

import java.util.Comparator;

import java.util.PriorityQueue;

import java.util.Scanner;

class Huffman {

public static void printCode(HuffmanNode root, String s)

{

//checks if the current node is a leaf node

if (root.left == null && root.right == null

&& Character.isLetter(root.c)) {

System.out.println(root.c + ":" + s);

return;

}

printCode(root.left, s + "0");

printCode(root.right, s + "1");

}

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

int n = 6;

char[] charArray = { 'a', 'b', 'c', 'd', 'e', 'f' };

int[] charfreq = { 5, 9, 12, 13, 16, 45 };

PriorityQueue<HuffmanNode> q

= new PriorityQueue<HuffmanNode>(

n, new MyComparator());

for (int i = 0; i < n; i++) {

HuffmanNode hn = new HuffmanNode();

hn.c = charArray[i];

hn.data = charfreq[i];

hn.left = null;

hn.right = null;

q.add(hn);

}

HuffmanNode root = null;

while (q.size() > 1) {

HuffmanNode x = q.peek();

q.poll();

HuffmanNode y = q.peek();

q.poll();

HuffmanNode f = new HuffmanNode();

f.data = x.data + y.data;

f.c = '-';

f.left = x;

f.right = y;

root = f;

q.add(f);

}

printCode(root, "");

}

}

class HuffmanNode {

int data;

char c;

HuffmanNode left;

HuffmanNode right;

}

class MyComparator implements Comparator<HuffmanNode> {

public int compare(HuffmanNode x, HuffmanNode y)

{

return x.data - y.data;

}

}