

```

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.FileNotFoundException;
public class Codificador {

    static BufferedReader entrada;

    public static void main (String[] arEntrada) throws IOException {

        //Recebe entradas

        if ( arEntrada.length > 0 ) {
            try {
                entrada = new BufferedReader ( new FileReader ( arEntrada[0] ) );
            }
            catch (FileNotFoundException erro) {
                System.out.println ("" );
                System.out.println (erro);
                System.out.println ("Arquivo nao encontrado, tente novamente");
                System.out.println ("" );
                System.exit(0);
            }
        }
        else {
            System.out.println ("Entre com um arquivo na linha de comando");
            System.exit (0);
        }

        //Monta as entradas

        HufPar[] entradas = new HufPar[ Integer.parseInt(entrada.readLine()) ];

        String[] linha = new String[2];
        for ( int i=0; i<entradas.length;i++ ) {
            linha = entrada.readLine().split("\\s");
            char c = linha[0].charAt(0);
            double n = Double.parseDouble ( linha[1] );

            entradas[i] = new HufPar ( c, n );
        }

        //Inicia a construcao da arvore

        HufEncoder coder = new HufEncoder ( entradas );
        NodoBinario codificacao = coder.geraArvoreDeHuffman();
        ImprimeArvore (codificacao,"");

    }
    static void ImprimeArvore(NodoBinario n, String str) {
        if ( (n.getEsquerdo()==null) && (n.getDireito()==null) ) {
            System.out.println(n.getHufPar().obtemCaracter()+"\t"+str);
        } else {
            ImprimeArvore(n.getEsquerdo(), str+"0");
            ImprimeArvore(n.getDireito(), str+"1");
        }
    }
}

```