

```
1  /* Problem 3--Domino Chains
2     This program does nothing more than basic trial and error. */
3
4  import java.io.*;
5  import java.util.*;
6
7  public class prob3 {
8
9     private static Scanner in;
10    private static PrintWriter out;
11    private static int cs;
12
13    public static void main (String[] args) throws Exception {
14
15        in = new Scanner (new File ("prob3.in"));
16        out = new PrintWriter ("prob3.out");
17        cs = 1;
18        int sz;
19        while ((sz = in.nextInt ()) > 0) { //Read in the dominoes.
20            int [][] D = new int[sz][2];
21            for (int i=0; i < sz; i++) {
22                D[i][0] = in.nextInt(); D[i][1] = in.nextInt();
23            }
24            out.print ("Case "+(cs++)+": The longest chain contains "+
25                Process (new int[0][2],D)+" dominoes.\r\n\r\n");
26        }
27        in.close ();
28        out.close ();
29    }
30
31    /* Process processes each case and computes and returns the
32       maximum length.  chain contains a chain discovered so far.  rest
33       is the rest of the dominoes that may or may not fit in the chain. */
34    public static int Process (int[][] chain, int[][] rest)
35        throws Exception {
36
37        if (rest.length==0) return chain.length; //They're all in the chain
38        int maxl = chain.length; //We already have a bunch in a chain.
39        for (int i=0; i < rest.length; i++)
40            if (chain.length==0 || chain[chain.length-1][1] == rest[i][0]) {
41                //Check the rest of the dominoes to add to the chain
42                int[][] newchain = new int[chain.length+1][2];
43                int[][] newrest = new int[rest.length-1][2];
44                for (int j=0; j < chain.length; j++) newchain[j] = chain[j];
45                newchain[chain.length] = rest[i]; //add it to the chain
46                for (int j=0, k=0; j < rest.length; j++) {
47                    if (i==j) continue;
48                    newrest[k++] = rest[j];
49                } //Recursively process the rest of the chain.
50                maxl = Math.max (Process (newchain,newrest),maxl);
51                if (maxl==chain.length+rest.length) return maxl;
52            }
53        return maxl;
54    }
55 }
56
```