```
 1  /* Problem 6--Leap Frog
 2     This is actually a classic puzzle that you can buy at a puzzle store.
 3     It's kind of neat that the solution is so nicely patterned. */
 4
 5  import java.io.*;
 6  import java.util.*;
 7
 8  public class prob6 {
 9
10   private static Scanner in = null;
11   private static PrintWriter out = null;
12   private static int cs = 0, sp = 0;
13   private static char[] lf = null;
14
15   public static void main (String[] args) throws Exception {
16
17    in = new Scanner (new File ("prob6.in"));
18    out = new PrintWriter ("prob6.out");
19    while (true) {
20     int sz = in.nextInt ();
21     if (sz==0) break;
22     Process (sz); //Process each data case
23    }
24    in.close ();
25    out.close ();
26   }
27
28   /* Solves the problem with sz boys and sz girls */
29   public static void Process (int sz) {
30
31    out.print ("Case "+(++cs)+"\r\n\r\n");
32    lf = new char[2*sz+1];
33    for (int i=0; i < sz; i++) lf[i] = 'B'; //Build the first board
34    lf[sz] = ' ';
35    for (int i=0; i < sz; i++) lf[sz+1+i] = 'G';
36    sp = sz;
37    out.print (new String(lf)+"\r\n"); //Print the first board
38    while (true) {
39     if (MoveB()) break; //Move boys
40     if (MoveG()) break; //Move girls
41    }
42    out.print ("\r\n");
43   }
44
45   /* MoveB moves the boys.  The strategy is this:  a move is based
46      entirely on jumps, except for possibly the first or last move.
47      If the first move is a straight move, then the last move cannot
48      be. This returns true if the problem is already soled. */
49   public static boolean MoveB () {
50
51    int rb = 0;
52    for (;rb < lf.length;rb++)
53     if (lf[rb]=='B') break; //If the first boy comes after the space,
54    if (sp < rb) return true; //DONE!
55    boolean later=true;
56    if (lf[sp-1]=='B') { //make a direct move if we can
57     lf[sp] = 'B'; lf[sp-1] = ' '; sp--;
58     later = false;
59     out.print (new String (lf)+"\r\n");
60    } //Start jumping
61    while (sp > 1 && lf[sp-1]=='G' && lf[sp-2]=='B') {
62     lf[sp] = 'B'; lf[sp-2] = ' '; sp-=2;
63     out.print (new String (lf) + "\r\n");
64    } //make a direct move if we can
65    if (later && sp > 0 && lf[sp-1]=='B') {
66     lf[sp] = 'B'; lf[sp-1] = ' '; sp--;
67     out.print (new String (lf)+"\r\n");
68    }
```

```
69    return false;
70   }
71
72   /* MoveG is just like MoveB, only as a mirror image. */
73   public static boolean MoveG () {
74
75    int rb = lf.length-1;
76    for (;rb >= 0; rb--)
77     if (lf[rb]=='G') break; //If the last girl comes before the space,
78    if (sp > rb) return true; //DONE!
79    boolean later=true;
80    if (lf[sp+1]=='G') { //make a direct move if we can
81     lf[sp] = 'G'; lf[sp+1] = ' '; sp++;
82     later = false;
83     out.print (new String (lf)+"\r\n");
84    } //start jumping
85    while (sp < lf.length-2 && lf[sp+1]=='B' && lf[sp+2]=='G') {
86     lf[sp] = 'G'; lf[sp+2] = ' '; sp+=2;
87     out.print (new String (lf) + "\r\n");
88    } //make a direct move if we can
89    if (later && sp < lf.length-1 && lf[sp+1]=='G') {
90     lf[sp] = 'G'; lf[sp+1] = ' '; sp++;
91     out.print (new String (lf)+"\r\n");
92    }
93    return false;
94   }
95  }
96
```