

```

29
30 public static boolean Recherche_des_Paires_Isolées() {
31
32     ArrayList<Case> liste = new ArrayList<>();
33
34     for (Case kase : liste_des_cases_vides) {
35         int couple = kase.getCouple();
36         if (couple != 0) {
37
38             if (Paire_isolée_dans_la_colonne(kase.colonne, couple, liste)) {
39                 Afficher_la_paire(couple, liste);
40                 return true;
41             }
42
43             if (Paire_isolée_dans_la_ligne(kase.ligne, couple, liste)) {
44                 Afficher_la_paire(couple, liste);
45                 return true;
46             }
47
48             if (Paire_isolée_dans_le_bloc(kase.bloc, couple, liste)) {
49                 Afficher_la_paire(couple, liste);
50                 return true;
51             }
52         }
53     }
54
55     return false;
56 }
57
58 static boolean Paire_isolée_dans_la_colonne(int colonne, int couple, ArrayList<Case> liste) {
59
60     if (Compte_du_couple_dans_la_colonne(colonne, couple, liste) == 2) {
61         if (Exclus_dans_la_colonne(colonne, couple, liste)) {
62             return true; // Paire isolée trouvée dans la colonne.
63         }
64     }
65     return false;
66 }
67
68 static int Compte_du_couple_dans_la_colonne(int colonne, int couple, ArrayList<Case> liste) {
69
70     liste.clear();
71     for (int ligne : de_1_à_9) {
72         Case case_scrutée = la_case(colonne + 9 * (ligne - 1));
73         if (case_scrutée.getCouple() == couple) {
74             liste.add(case_scrutée);
75         }
76     }
77     return liste.size();
78 }
79
80 static boolean Exclus_dans_la_colonne(int colonne, int couple, ArrayList<Case> liste) {
81
82     int candidat1 = couple / 10;
83     int candidat2 = couple % 10;
84     liste_des_candidats_exclus.clear();
85     for (int ligne : de_1_à_9) {
86         Case case_scrutée = la_case(colonne + 9 * (ligne - 1));
87         if (!liste.contains(case_scrutée)) {
88             if (case_scrutée.contient_le(candidat1)) {
89                 liste_des_candidats_exclus.add(candidat1 * 100 + colonne + 9 * (ligne - 1));
90             }
91             if (case_scrutée.contient_le(candidat2)) {
92                 liste_des_candidats_exclus.add(candidat2 * 100 + colonne + 9 * (ligne - 1));
93             }
94         }
95     }
96     if (!liste_des_candidats_exclus.isEmpty()) {
97         return true; // Exclus trouvés dans la colonne.
98     }

```

```

99     }
100     return false;
101 }
102
103 static boolean Paire_isolée_dans_la_ligne(int ligne, int couple, ArrayList<Case> liste) {
104
105     if (Compte_du_couple_dans_la_ligne(ligne, couple, liste) == 2) {
106         if (Exclus_dans_la_ligne(ligne, couple, liste)) {
107             return true; // Paire isolée trouvée dans la ligne.
108         }
109     }
110     return false;
111 }
112
113 static int Compte_du_couple_dans_la_ligne(int ligne, int couple, ArrayList<Case> liste) {
114
115     liste.clear();
116     for (int colonne : de_1_à_9) {
117         Case case_scrutée = la_case(colonne - 1 + 9 * ligne);
118         if (case_scrutée.getCouple() == couple) {
119             liste.add(case_scrutée);
120         }
121     }
122     return liste.size();
123 }
124
125 static boolean Exclus_dans_la_ligne(int ligne, int couple, ArrayList<Case> liste) {
126
127     int candidat1 = couple / 10;
128     int candidat2 = couple % 10;
129     liste_des_candidats_exclus.clear();
130     for (int colonne : de_1_à_9) {
131         Case case_scrutée = la_case(colonne - 1 + 9 * ligne);
132         if (!liste.contains(case_scrutée)) {
133             if (case_scrutée.contient_le(candidat1)) {
134                 liste_des_candidats_exclus.add(candidat1 * 100 + (colonne - 1) + 9 * ligne);
135             }
136             if (case_scrutée.contient_le(candidat2)) {
137                 liste_des_candidats_exclus.add(candidat2 * 100 + (colonne - 1) + 9 * ligne);
138             }
139         }
140     }
141     if (!liste_des_candidats_exclus.isEmpty()) {
142         return true; // Exclus trouvés dans la ligne.
143     }
144     return false;
145 }
146
147 static boolean Paire_isolée_dans_le_bloc(int bloc, int couple, ArrayList<Case> liste) {
148
149     if (Compte_du_couple_dans_le_bloc(bloc, couple, liste) == 2) {
150         if (Exclus_dans_le_bloc(bloc, couple, liste)) {
151             return true; // Paire isolée trouvée dans le bloc.
152         }
153     }
154     return false;
155 }
156
157 static int Compte_du_couple_dans_le_bloc(int bloc, int couple, ArrayList<Case> liste) {
158
159     liste.clear();
160     for (int case_de_bloc : de_1_à_9) {
161         Case case_scrutée = la_case(3 * (bloc % 3) + (case_de_bloc - 1) % 3
162             + 9 * (3 * (bloc / 3) + (case_de_bloc - 1) / 3));
163         if (case_scrutée.getCouple() == couple) {
164             liste.add(case_scrutée);
165         }
166     }
167     return liste.size();
168 }

```

```
169
170 static boolean Exclus_dans_le_bloc(int bloc, int couple, ArrayList<Case> liste) {
171
172     int candidat1 = couple / 10;
173     int candidat2 = couple % 10;
174     liste_des_candidats_exclus.clear();
175     for (int case_de_bloc : de_1_à_9) {
176         Case case_scrutée = la_case(3 * (bloc % 3) + (case_de_bloc - 1) % 3
177             + 9 * (3 * (bloc / 3) + (case_de_bloc - 1) / 3));
178         if (!liste.contains(case_scrutée)) {
179             if (case_scrutée.contient_le(candidat1)) {
180                 liste_des_candidats_exclus.add(candidat1 * 100
181                     + case_scrutée.colonne + 9 * case_scrutée.ligne);
182             }
183             if (case_scrutée.contient_le(candidat2)) {
184                 liste_des_candidats_exclus.add(candidat2 * 100
185                     + case_scrutée.colonne + 9 * case_scrutée.ligne);
186             }
187         }
188     }
189     if (!liste_des_candidats_exclus.isEmpty()) {
190         return true; // Exclus trouvés dans le bloc.
191     }
192     return false;
193 }
194
195 /*-----*/
196
```