

```

1  /*
2  -----
3  Laboratoire : 02_A
4  Fichier      : exercice_2.cpp
5  Auteur(s)   : Thomas Ricci, Eric Bousbaa, Thibaud Franchetti
6  Date        : 17.10.2018
7
8  But          : Demande à l'utilisateur d'entrer un numéro de case d'un damier
9  *            de hauteur et largeur déterminées puis affiche les numéros des cases
10 *            des symétries suivantes :
11 *              - Symétrie verticale
12 *              - Symétrie horizontale
13 *              - Symétrie diagonale gauche
14 *              - Symétrie diagonale droite.
15
16 Remarque(s) : - Aucun contrôle sur la saisie utilisateur n'est fait.
17 *              - Le côté du damier peut être pair ou impair.
18 *              - Les diagonales ne faisant pas partie des axes de symétrie d'un
19 *                rectangle, le damier doit être carré.
20
21 Compilateur : - g++ (Ubuntu 5.4.0-6ubuntu1~16.04.10),
22 *              - Apple LLVM version 9.0.0 (clang-900.0.39.2),
23 *              - g++ (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0
24 -----
25 */
26
27 #include <cstdlib>
28 #include <iostream>
29
30 using namespace std;
31
32 int main(){
33
34     const int COTE = 8;
35     int nombre;
36
37     cout << "Choisissez un numero de case entre 1 et " << COTE * COTE << " : ";
38     cin >> nombre;
39
40     // Passage d'une numérotation "linéaire" des cases à un système de coordonnées
41     // dont l'origine est située sur la case en haut à gauche du damier.
42     int nombre_x = (nombre - 1) % COTE;
43     int nombre_y = (nombre - 1) / COTE;
44
45     // Calcul des coordonnées de la case symétrique par rapport à l'axe horizontal.
46     int sym_horizontale_y = COTE - nombre_y - 1;
47     int sym_horizontale_x = nombre_x;
48     // Les coordonnées sont reconverties en un numéro de case avant d'être affichées.
49     cout << "Case symetrique par rapport a l'axe horizontal      : "
50          << sym_horizontale_y * COTE + sym_horizontale_x + 1
51          << endl;
52
53     // Calcul des coordonnées de la case symétrique par rapport à l'axe vertical.
54     int sym_verticale_y = nombre_y;
55     int sym_verticale_x = COTE - nombre_x - 1;
56     cout << "Case symetrique par rapport a l'axe vertical        : "
57          << sym_verticale_y * COTE + sym_verticale_x + 1
58          << endl;
59
60     // Calcul des coordonnées de la case symétrique par rapport à la diagonale droite.
61     int sym_diag_droite_x = COTE - nombre_y - 1;
62     int sym_diag_droite_y = COTE - nombre_x - 1;
63     cout << "Case symetrique par rapport a la diagonale droite : "
64          << sym_diag_droite_y * COTE + sym_diag_droite_x + 1
65          << endl;
66
67     // Calcul des coordonnées de la case symétrique par rapport à la diagonale gauche.
68     cout << "Case symetrique par rapport a la diagonale gauche : "
69          << nombre_x * COTE + nombre_y + 1
70          << endl;
71
72     cout << "Appuyez sur une touche pour continuer..." << endl;
73     cin.get();
74
75     return EXIT_SUCCESS;
76 }
77

```