

```

1  /*
2  -----
3  Fichier      : calendrier.cpp
4  Nom du labo  : Laboratoire 03
5  Auteur(s)    : Flavio Sovilla, Nicolas Benitez
6  Date         : 26.10.2020
7  But          : Permettre à l'utilisateur d'entrer une date entre 1900
8                  et 2100. Le programme va par la suite ressortir le
9                  calendrier de l'année avec tous les mois et les dates
10                 des différents jours.
11
12  Remarque(s) : Les valeurs de verifications d'une année bissextile ont été laissées
13                  en dur dans le code puisqu'elles proviennent d'une formule. C'est aussi
14                  le cas pour la formule du calcul du bon jour du mois de janvier.
15
16                  Source pour calcul du jour du 1er Janvier :
17                  https://cs.uwaterloo.ca/~alopez-o/math-faq/node73.html
18
19                  Source pour calcul du nombre de jours dans 1 mois :
20                  http://www.dispersiondesign.com/articles/time/number\_of\_days\_in\_a\_month
21
22  Compilateur : MinGW-w64
23  -----
24  */
25  #include <cstdlib> // Pour le EXIT_SUCCESS
26  #include <iostream> // Pour le flux
27  #include <limits> // Pour le numeric_limits<streamsize>
28  #include <iomanip> // Pour le setw(...)
29  #include <cmath> // Pour le calcul du jour du 1er Janvier -> fonction floor()
30
31  using namespace std;
32
33  #define VIDER_BUFFER cin.ignore(numeric_limits<streamsize>::max(), '\n')
34
35  int main() {
36
37      // -----
38      // Variables constantes
39      const int ESPACE_JOURS = 2,
40                ESPACE_ALIGNEMENT = 3,
41                NBRE_COLONNES = 7,
42                ANNEE_MIN = 1900,
43                ANNEE_MAX = 2100,
44                MOIS_DEPART = 1,
45                MOIS_FIN = 12;
46
47      // -----
48      // Variables générales
49      enum class Mois {
50          JANVIER = 1,
51          FEVRIER,
52          MARS,
53          AVRIL,
54          MAI,
55          JUIN,
56          JUILLET,
57          AOUT,
58          SEPTEMBRE,
59          OCTOBRE,
60          NOVEMBRE,
61          DECEMBRE
62      };
63
64      char choixUtilisateur;
65
66      // -----
67      // Message de bienvenue
68      cout << "ce programme ..." << endl;
69
70      // -----
71      // Boucle de recommencement
72      do{
73
74          // -----
75          // Variables utilisées pour gérer l'entrée de l'utilisateur
76          int annee;
77

```

```

78     bool verifAnnee = true,
79         bissextile = true;
80
81     // -----
82     // Boucle Saisie utilisateur pour l'annee
83     do {
84         cout << "entrer une valeur [" << ANNEE_MIN << "-" << ANNEE_MAX << "] : ";
85         cin >> annee;
86
87         if (cin.fail()) {
88             cin.clear();
89         }
90         VIDER_BUFFER;
91
92         if (annee >= ANNEE_MIN && annee <= ANNEE_MAX) {
93             verifAnnee = false;
94         }
95         else {
96             cout << "!!\n recommencer" << endl;
97         }
98     } while (verifAnnee);
99     //FIN - Boucle Saisie utilisateur pour l'annee
100
101     cout << endl;
102
103     // -----
104     // Vérification de l'année bissextile
105     // Ne pas modifier les valeurs (voir remarques) !
106     bissextile = (annee % 400 == 0 || (annee % 4 == 0 && annee % 100 != 0));
107
108     // -----
109     // Formule pour calculer le jour de la semaine pour le 1er janvier
110     // Ne pas modifier les valeurs (voir remarques) !
111     int decennie = (annee % 100) - 1 ; // les deux derniers chiffres de l'annee
112     int siecle = annee / 100; // les deux premiers chiffres de l'annee
113     int premierJanvier = (1
114         + (int)floor((2.6 * 11) - 0.2)
115         + (int)floor(decennie / 4)
116         + (int)floor(siecle / 4)
117         - 2 * siecle + decennie)
118         % 7;
119
120     // Un decalage s'opere sur le 1er Janvier puisque le resultat du calcul precedent
121     // se base sur une semaine commençant un dimanche
122     // donc Dimanche = 0, Lundi = 1, Mardi = 2, etc...
123     int jourDepart =
124         premierJanvier == 0 ?
125         premierJanvier + 6 :
126         premierJanvier - 1 ;
127
128     // -----
129     // Boucle d'affichage du calendrier
130     for (int i = MOIS_DEPART; i <= MOIS_FIN; ++i) {
131
132         int nbreJour = 1;
133
134         switch(i) {
135             case (int)Mois::JANVIER: cout << "JANVIER ";
136                                     break;
137             case (int)Mois::FEVRIER: cout << "FEVRIER ";
138                                     break;
139             case (int)Mois::MARS: cout << "MARS ";
140                                   break;
141             case (int)Mois::AVRIL: cout << "AVRIL ";
142                                   break;
143             case (int)Mois::MAI: cout << "MAI ";
144                                   break;
145             case (int)Mois::JUIN: cout << "JUIN ";
146                                   break;
147             case (int)Mois::JUILLET: cout << "JUILLET ";
148                                     break;
149             case (int)Mois::AOUT: cout << "AOUT ";
150                                   break;
151             case (int)Mois::SEPTEMBRE: cout << "SEPTEMBRE ";
152                                     break;
153             case (int)Mois::OCTOBRE: cout << "OCTOBRE ";
154                                     break;

```

```

155         case (int)Mois::NOVEMBRE: cout << "NOVEMBRE ";
156                                     break;
157         case (int)Mois::DECEMBRE:  cout << "DECEMBRE ";
158                                     break;
159         default: break;
160     }
161
162     // -----
163     // Calcul du nombre de jours d'un mois
164     // On remarque un pattern qui se répète tous les 7 mois, dans lequel se trouve
165     // un deuxième pattern se repetant tous les 2 mois. Il ne reste plus qu'a gerer
166     // le mois de fevrier et les annees bissextiles. Cela donne le calcul suivant
167     int nbreJoursMois = (i == 2) ? (28 + bissextile) : 31 - (i - 1) % 7 % 2;
168
169
170     // -----
171     // Affichage des colonnes avec gestion des espaces pour le jour de depart du
172     // mois en cours
173     cout << annee << endl;
174     cout << " L M M J V S D" << endl;
175
176     int sautJours = 0;
177
178     while(jourDepart--) {
179         cout << setw(ESPACE_ALIGNEMENT) << " ";
180
181         ++sautJours;
182
183         if (sautJours % NBRE_COLONNES == 0) {
184             cout << endl;
185         }
186     }
187
188     // -----
189     // Affichage des jours du mois
190     while(nbreJour <= nbreJoursMois){
191         cout << setw(ESPACE_JOURS) << nbreJour << " ";
192
193         ++nbreJour;
194         ++sautJours;
195
196         if (sautJours % NBRE_COLONNES == 0) {
197             cout << endl;
198         }
199     }
200     cout << endl;
201
202     // On recalcule le jour de depart pour le mois suivant
203     jourDepart = sautJours % NBRE_COLONNES;
204
205     // On met un saut de ligne s'il n'y en a pas deja un
206     if (jourDepart) {
207         cout << endl;
208     }
209 } //FIN - Boucle d'affichage du calendrier
210
211 // -----
212 // Boucle Saisie utilisateur pour recommencer avec gestion des entrees erronees
213 do{
214     cout << "Voulez-vous recommencer [O/N] ? ";
215     cin >> choixUtilisateur;
216
217     if (cin.fail()) {
218         cin.clear();
219     }
220     VIDER_BUFFER;
221
222 }while(choixUtilisateur != 'N' && choixUtilisateur != 'O');
223 //FIN - Boucle Saisie utilisateur pour recommencer
224
225 } while(choixUtilisateur == 'O');
226 //FIN - Boucle de recommencement
227
228 return EXIT_SUCCESS;
229 }

```