

```

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 << "//!\\ 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     cout << "Bissextile : " << bissextile << endl;
108
109     // -----
110     // Formule pour calculer le jour de la semaine pour le 1er janvier
111     // Ne pas modifier les valeurs (voir remarques) !
112     int decennie = (annee % 100) - 1 ; // les deux derniers chiffres de l'annee
113     int siecle = annee / 100; // les deux premiers chiffres de l'annee
114     int premierJanvier = (1
115         + (int)floor((2.6 * 11) - 0.2)
116         + (int)floor(decennie / 4)
117         + (int)floor(siecle / 4)
118         - 2 * siecle + decennie)
119         % 7;
120
121     // Certains resultats peuvent etre negatifs et fausser le jour, c'est pourquoi
122     // nous devons ajouter cette ligne dans le but de parer cette eventualiter
123     premierJanvier = premierJanvier < 0 ? premierJanvier + 7 : premierJanvier;
124
125     // Un decalage s'opere sur le 1er Janvier puisque le resultat du calcul precedent
126     // se base sur une semaine commençant un dimanche
127     // donc Dimanche = 0, Lundi = 1, Mardi = 2, etc...
128     int jourDepart =
129         premierJanvier == 0 ?
130         premierJanvier + 6 :
131         premierJanvier - 1 ;
132
133     // -----
134     // Boucle d'affichage du calendrier
135     for (int i = MOIS_DEPART; i <= MOIS_FIN; ++i) {
136
137         int nbreJour = 1;
138
139         switch(i) {
140             case (int)Mois::JANVIER: cout << "JANVIER ";
141                                     break;
142             case (int)Mois::FEVRIER: cout << "FEVRIER ";
143                                     break;
144             case (int)Mois::MARS: cout << "MARS ";
145                                     break;
146             case (int)Mois::AVRIL: cout << "AVRIL ";
147                                     break;
148             case (int)Mois::MAI: cout << "MAI ";
149                                     break;
150             case (int)Mois::JUIN: cout << "JUIN ";
151                                     break;
152             case (int)Mois::JUILLET: cout << "JUILLET ";
153                                     break;
154             case (int)Mois::AOUT: cout << "AOUT ";
155                                     break;

```

```

155         case (int)Mois::SEPTEMBRE: cout << "SEPTEMBRE ";
156                                     break;
157         case (int)Mois::OCTOBRE:   cout << "OCTOBRE ";
158                                     break;
159         case (int)Mois::NOVEMBRE:  cout << "NOVEMBRE ";
160                                     break;
161         case (int)Mois::DECEMBRE:  cout << "DECEMBRE ";
162                                     break;
163         default: break;
164     }
165
166     // -----
167     // Calcul du nombre de jours d'un mois
168     // On remarque un pattern qui se répète tous les 7 mois, dans lequel se trouve
169     // un deuxième pattern se repetant tous les 2 mois. Il ne reste plus qu'a gerer
170     // le mois de fevrier et les anneess bissextiles. Cela donne le calcul suivant
171     int nbreJoursMois = (i == 2) ? (28 + bissextile) : 31 - (i - 1) % 7 % 2;
172
173     // -----
174     // Affichage des colonnes avec gestion des espaces pour le jour de depart du
175     // mois en cours
176     cout << annee << endl;
177     cout << " L M M J V S D" << endl;
178
179     int sautJours = 0;
180
181     while(jourDepart--) {
182         cout << setw(ESPACE_ALIGNEMENT) << " ";
183
184         ++sautJours;
185
186         if (sautJours % NBRE_COLONNES == 0) {
187             cout << endl;
188         }
189     }
190
191     // -----
192     // Affichage des jours du mois
193     while(nbreJour <= nbreJoursMois){
194         cout << setw(ESPACE_JOURS) << nbreJour << " ";
195
196         ++nbreJour;
197         ++sautJours;
198
199         if (sautJours % NBRE_COLONNES == 0) {
200             cout << endl;
201         }
202     }
203     cout << endl;
204
205     // On recalcule le jour de depart pour le mois suivant
206     jourDepart = sautJours % NBRE_COLONNES;
207
208     // On met un saut de ligne s'il n'y en a pas deja un
209     if (jourDepart) {
210         cout << endl;
211     }
212 } //FIN - Boucle d'affichage du calendrier
213
214 // -----
215 // Boucle Saisie utilisateur pour recommencer avec gestion des entrees erronees
216 do{
217     cout << "Voulez-vous recommencer [O/N] ? ";
218     cin >> choixUtilisateur;
219
220     if (cin.fail()) {
221         cin.clear();
222     }
223     VIDER_BUFFER;
224
225     while(choixUtilisateur != 'N' && choixUtilisateur != 'O');
226 }while(choixUtilisateur != 'N' && choixUtilisateur != 'O');
227 //FIN - Boucle Saisie utilisateur pour recommencer
228
229 } while(choixUtilisateur == 'O');
230 //FIN - Boucle de recommencement
231

```

```
232     return EXIT_SUCCESS;
233 }
```