

```
/*
-----
Fichier      : calendrier.cpp
Nom du labo  : Labo 03 - Calendrier
Auteur(s)    : Richard Tenorio, Damiano Mondaini
Date         : 27.10.2020
But          : Le programme a pour but d'afficher le calendrier d'une année
              selon l'entrée de l'utilisateur. La mise en page du calendrier
              est soignée.

Remarque(s) : - L'utilisateur ne peut entrer que des années entre 1900 et
              2100
              - Les saisies de l'utilisateur sont vérifiées

Compilateur  : Mingw-w64 g++ 8.1.0
-----
*/

#include <iostream> // Gestion du flux
#include <iomanip>   // Gestion de l'affichage
#include <cstdlib>   // EXIT_SUCCESS
#include <limits>    // numeric_limits<...> pour vider le buffer
#include <string>    // Type de variable string
#include <cmath>     // Utilisation de floor()

using namespace std;

// Définitions des commandes pour gérer le buffer et afficher le message de fin
#define VIDER_BUFFER    cin.ignore(numeric_limits<streamsize>::max(), '\n')
#define MESSAGE_FIN     cout << "Presser ENTER pour quitter"; \
                        VIDER_BUFFER
#define REPARER_BUFFER  cin.clear()

int main () {

    // Accueil du programme
    cout << "Ce programme permet d'afficher le calendrier correspondant à une année" << endl;

    string recommencerStatut; // Variable permettant de savoir si l'utilisateur veut recommencer

    // Exécute une fois le programme et recommence si l'utilisateur le souhaite
    do {

        // Déclaration et initialisation des variables et constantes pour gérer les années
        const unsigned int anneeMax    = 2100;
        const unsigned int anneeMin    = 1900;
        const unsigned int noEspaceJour = 2;
        unsigned int anneeSaisie      = 0;

        // Saisi et vérification de l'année entrée par l'utilisateur
        do {
            cout << "Entrer une valeur [1900-2100] : ";
```

```
cin >> anneeSaisie;

if (cin.fail() || anneeSaisie < anneeMin || anneeSaisie > anneeMax) {
    REPARER_BUFFER;
    cout << "La valeur saisie n'est pas valide. Merci de recommencer." << endl;
}
VIDER_BUFFER;
} while (anneeSaisie < anneeMin || anneeSaisie > anneeMax);

// Déclaration de class enum Mois avec janvier qui commence à 1
enum class Mois {
    JANVIER = 1,
    FEVRIER,
    MARS,
    AVRIL,
    MAI,
    JUIN,
    JUILLET,
    AOUT,
    SEPTEMBRE,
    OCTOBRE,
    NOVEMBRE,
    DECEMBRE
};

// Déclaration de class enum Jour avec lundi qui commence à 1
enum class Jours {
    LUNDI = 1,
    MARDI,
    MERCREDI,
    JEUDI,
    VENDREDI,
    SAMEDI,
    DIMANCHE
};

// Définition, initialisation et calcul si l'année est bissextile
bool estBissextile = false;
if (anneeSaisie % 400 == 0 || anneeSaisie % 4 == 0 && anneeSaisie % 100 != 0) {
    estBissextile = true;
}

// Declaration du premier jour semaine et premier jour du mois
Jours premierJourMois;
Jours jourMois = Jours::LUNDI;

// Premier jour de la semaine de janvier de l'année saisi
// Source : https://www.tutorialspoint.com/day-of-the-week-in-cplusplus
unsigned int deuxDerniersChiffres = (anneeSaisie - 1) % 100;
unsigned int deuxPremiersChiffres = (anneeSaisie - 1) / 100;
unsigned int codeJourSemaine
    = (1 + (int)floor((13*14)/5)
        + deuxDerniersChiffres + (int)floor(deuxDerniersChiffres / 4)
        + (int)floor(deuxPremiersChiffres / 4))
```

```
        + (5 * deuxPremiersChiffres));

// Décode 'codeJourSemaine': quand codeJourSemaine = 0, correspond à samedi et quand codeSemaine = 6
// correspond à vendredi
switch (codeJourSemaine % 7) {
    case 0:
        premierJourMois = Jours::SAMEDI;
        break;
    case 1:
        premierJourMois = Jours::DIMANCHE;
        break;
    case 2:
        premierJourMois = Jours::LUNDI;
        break;
    case 3:
        premierJourMois = Jours::MARDI;
        break;
    case 4:
        premierJourMois = Jours::MERCREDI;
        break;
    case 5:
        premierJourMois = Jours::JEUDI;
        break;
    case 6:
        premierJourMois = Jours::VENDREDI;
        break;
}

// Boucle pour chaque mois
for (Mois moisCourrant = Mois::JANVIER;
     moisCourrant <= Mois::DECEMBRE;
     moisCourrant = Mois((int)moisCourrant + 1)) {

    // Affiche le nom du mois en fonction de son numéro
    switch ((int)moisCourrant) {
        case 1: cout << "JANVIER"; break;
        case 2: cout << "FEVRIER"; break;
        case 3: cout << "MARS"; break;
        case 4: cout << "AVRIL"; break;
        case 5: cout << "MAI"; break;
        case 6: cout << "JUIN"; break;
        case 7: cout << "JUILLET"; break;
        case 8: cout << "AOUT"; break;
        case 9: cout << "SEPTEMBRE"; break;
        case 10: cout << "OCTOBRE"; break;
        case 11: cout << "NOVEMBRE"; break;
        case 12: cout << "DECEMBRE"; break;
    }

    // Affiche l'année
    cout << " " << anneeSaisie << endl;
```

```
// Affiche la premiere lettre du jour de la semaine
cout << " L M M J V S D" << endl;

int nbreJourMois;

// calcule le nombre de jour dans le mois courant
if (moisCourrant == Mois::FEVRIER) { // Vérifie si le mois courant est févirer
    if (estBissextile) {
        nbreJourMois = 29;
    } else {
        nbreJourMois = 28;
    }
} else if ((int)moisCourrant < (int)Mois::AOUT) { // Vérifie si le mois courant est avant aout
    if ((int)moisCourrant % 2 == 0) {
        nbreJourMois = 30;
    } else {
        nbreJourMois = 31;
    }
} else { // Vérifie si le mois courant est depuis août
    if ((int)moisCourrant % 2 == 0) {
        nbreJourMois = 31;
    } else {
        nbreJourMois = 30;
    }
}

// Affiche les espaces pour commencer le mois le bon jour de la semaine
for (; jourMois < premierJourMois; jourMois = Jours((int)jourMois + 1)) {
    cout << " ";
}

// Afficher le numéro du jour par rapport au mois
for (int dateJour = 1; dateJour <= nbreJourMois; ++dateJour) {
    cout << setw(noEspaceJour) << dateJour << " ";

    if (jourMois == Jours::DIMANCHE) {
        cout << endl;
        jourMois = Jours::LUNDI;
    } else {
        jourMois = Jours((int)jourMois + 1);
    }
}

cout << endl << endl;

// Calcule le premier jour du mois suivant
premierJourMois = jourMois;
jourMois = Jours::LUNDI;
}

// Vérifie si l'utilisateur veut recommencer
do {
    cout << "Voulez vous recommencer ? [o/n] : ";
```

```
    cin >> recommencerStatut;

    // Vérifie si le buffer est cassé
    if (cin.fail()) {
        REPARER_BUFFER;
    }

    // Vérifie si l'utilisateur ne plus recommencer
    if (recommencerStatut == "n") {
        VIDER_BUFFER;
        MESSAGE_FIN;
        return EXIT_SUCCESS;
    }

    // Vérifie si l'utilisateur a saisi autre chose que 'o' ou 'n'
    if (recommencerStatut != "o") {
        cout << "La valeur saisie n'est pas valide. Merci de recommencer." << endl;
    }

    VIDER_BUFFER;

    } while (recommencerStatut != "o");

} while (recommencerStatut == "o");

return EXIT_SUCCESS;
}
```