

```

/*
-----
Laboratoire : Laboratoire n°5 - Calendrier -> Laboratoire n°5b
Fichier      : Labo_05_Farina_Naimi.cpp
Auteurs      : Jimmy Farina, Zied Naimi
Correcteurs  : Julien Biefer et Sven Rouvinez
Date         : 15.11.2016

But          : Affiche le calendrier complet dans un intervalle choisi par
               l'utilisateur entre [1900 - 2100] en tenant compte des années
               bissextiles et en affichant les numéros de semaine selon la norme
               ISO 8601. Propose à l'utilisateur de relancer le programme.

Remarque(s) : Les fonctions ne tiennent pas compte des erreurs.

Compilateur  : Apple LLVM 8.0
               gcc version 6.2.1 20160916 (Red Hat 6.2.1-2) (GCC)
-----
*/

#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <limits>

#define VIDER_BUFFER cin.ignore(numeric_limits<streamsize>::max(), '\n')

using namespace std;

//=====
// Prototypes
//=====

// Affiche un mois d'une année donnée ainsi que la numérotation des semaines
void afficherMois(const int mois, const int annee, int& debutDuMois, int& numSemaine);

// Gère la saisie d'un nombre entier par l'utilisateur
int saisieInt(const string messageSaisie, const int borneMin, const int borneMax);

// Gère le recommencement du programme
bool saisieRecommencer(const char valeurVraieParam, const char valeurFausseParam);

// Vérifie si le début de l'intervalle moisDebut/anneeDebut et bien avant
// la fin de l'intervalle moisFin/anneeFin
// (ou le même mois sur la même année => on affiche que le mois)
bool verifieIntervalle(const int moisDebut, const int anneeDebut,
                      const int moisFin, const int anneeFin);

// Retourne le nombre de jours dans un mois
int nbrJoursMois(const int mois, const int annee);

// Défini si une année est bissextile pour une année donnée
bool estBissextile(const int anneeUtilisateur);

// Affiche une série de signe selon un symbole donné
void afficheBarre(char symbole, const char symboleEncadre, const int nbrSymbole);

// Retourne la première lettre d'un jour de la semaine
char intJourEnChar(const int numJourSemaine);

// Retourne le nom du mois par rapport à un chiffre
string intMoisEnString(const int numMois);

// Permet de contrôler et corriger le flux si nécessaire
bool controleFlux(const bool saisie);

// Renvoie le jour de la semaine (1 (lundi) et 7 (dimanche)) pour une date donnée
// Tiré de http://mathforum.org/library/drmath/view/55837.html
int jourSemaine(const int jour, const int mois, const int annee);

// Renvoie le nombre total de jours avant le début de moisFinal sur une année donnée
int totalJourMois(const int moisFinal, const int annee);

// Renvoie le n° de la semaine pour une date donnée (peu précis)
int numeroSemainePourDate(const int jour, const int mois, const int annee);

// Renvoie le numéro de la première semaine d'un mois en fonction du premier jour
// Implémenté selon ISO 8601 : https://fr.wikipedia.org/wiki/ISO\_8601
// Cette fonction ajuste la précédente
void ajusterNumeroPremiereSemaine(int &premiereSemaine, const int mois, const int annee);

//=====
// Programme principal
//=====

int main() {
    // Constantes et variables utilisées
    //=====
    const int ANNEE_MIN = 1900;
    const int ANNEE_MAX = 2100;
    const int MOIS_MIN = 1;
    const int MOIS_MAX = 12;
    const char RECOMMENCER_VRAI = 'o';

```

```

const char RECOMMENCER_FAUX = 'n';

bool intervalleOK;
int premierJourIntervalle = 1; // 0 correspond au lundi
int saisieMoisDebut,
    saisieMoisFin;
int saisieAnneeDebut,
    saisieAnneeFin;

// Boucle qui permet de reset le programme
do {
    cout << "Bonjour, ce programme vous permet d'afficher le calendrier complet "
           "d'un intervalle de temps en tenant compte des annees bissextiles."
           << endl << endl;

    // Saisie de l'intervalle + vérification
    //=====
    do {
        intervalleOK = true;

        cout << "Veuillez saisir le début de l'intervalle : " << endl;
        saisieMoisDebut = saisieInt("    Mois ["
                                   + to_string(MOIS_MIN) + "-" + to_string(MOIS_MAX)
                                   + "] : ", MOIS_MIN, MOIS_MAX);
        saisieAnneeDebut = saisieInt("    Annee ["
                                   + to_string(ANNEE_MIN) + "-" + to_string(ANNEE_MAX)
                                   + "] : ", ANNEE_MIN, ANNEE_MAX);

        cout << "Veuillez saisir la fin de l'intervalle : " << endl;
        saisieMoisFin = saisieInt("    Mois ["
                                   + to_string(MOIS_MIN) + "-" + to_string(MOIS_MAX)
                                   + "] : ", MOIS_MIN, MOIS_MAX);
        saisieAnneeFin = saisieInt("    Annee ["
                                   + to_string(saisieAnneeDebut) + "-" + to_string(ANNEE_MAX)
                                   + "] : ", saisieAnneeDebut, ANNEE_MAX);

        if (!verifieIntervalle(saisieMoisDebut, saisieAnneeDebut, saisieMoisFin, saisieAnneeFin)) {
            cout << "L'intervalle saisi n'est pas valide.";
            intervalleOK = false;
        }
        cout << endl << endl;
    } while(!intervalleOK);

    // Définition du premier jour de l'intervalle
    //=====
    premierJourIntervalle = jourSemaine(1, saisieMoisDebut, saisieAnneeDebut);

    // Affichage
    //=====
    // Pour boucler sur les années, on définit la première année et le premier mois
    // de l'intervalle, le premier jour de ce mois
    for (int annee = saisieAnneeDebut, moisDebut = saisieMoisDebut, numSemaine = -1,
         moisFin = MOIS_MAX, premierJourMois = premierJourIntervalle;
         annee <= saisieAnneeFin; ++annee) {

        // Si on a passé la première année, on veut afficher tous les mois
        if (annee == saisieAnneeDebut+1)
            moisDebut = MOIS_MIN;

        // Si on affiche la dernière année, il faut s'arrêter au dernier mois demandé
        if (annee == saisieAnneeFin)
            moisFin = saisieMoisFin;

        // Si janvier est la première valeur saisie, donc si numSemaine n'est pas modifiée
        // numSemaine possède toujours la valeur -1 qui permet d'afficher la semaine de
        // l'année précédente si nécessaire
        ajusterNumeroPremiereSemaine(numSemaine, moisDebut, annee);

        // On affiche les mois pour l'année en cours
        for (int mois = moisDebut; mois <= moisFin; ++mois) {
            afficherMois(mois, annee, premierJourMois, numSemaine);
            cout << endl << endl;
        }
    }

} while (saisieRecommencer(RECOMMENCER_VRAI, RECOMMENCER_FAUX));

// Fin du programme
//=====
cout << "Merci d'avoir utilise le programme." << endl
     << "Pressez sur ENTER pour quitter le programme..." << endl;
VIDER_BUFFER;

return EXIT_SUCCESS;
}

//=====
// Fonctions
//=====

void afficherMois(const int mois, const int annee, int& debutDuMois, int& numSemaine) {

```

```

// Définition des variables
int nbrJours = nbrJoursMois(mois, annee);

// Définition des constantes
const int MOIS_JANVIER = 1;
const int MOIS_DECEMBRE = 12;
const int NBR_JOUR_SEMAINE = 7;
const int NOMBRE_LIMITE_REPORT_SEMAINE = 52;
const int ESPACE_NUMERO = 3;
const int ESPACE_SEMAINE = 3;
// Le + 1 correspond à l'espace entre le numéro de la semaine et la numérotation des jours
const int NBRE_SYM = NBR_JOUR_SEMAINE * ESPACE_NUMERO + ESPACE_SEMAINE + 1;
const char SYMBOLE = '=';
const char ENCADRE = '|';

cout << intMoisEnString(mois) << " " << annee << endl;
afficheBarre(SYMBOLE, ENCADRE, NBRE_SYM);

// On affiche les lettres des jours de la semaine
// Le + 1 et l'espace compensent les symboles encadrés dans la barre
cout << setw(ESPACE_SEMAINE + 1) << " ";
for (int jour = 1; jour <= NBR_JOUR_SEMAINE; ++jour)
    cout << setw(ESPACE_NUMERO) << intJourEnChar(jour);
cout << endl;

// On affiche les jours du mois dans les bonnes colonnes
// Ici jourMois débute à 1 puisqu'il est affiché
for (int jourMois = 1, compteur = 1; jourMois <= nbrJours; ++compteur) {
    // On affiche le n° de la semaine
    if (compteur % 7 == 1) {
        // SEMAINE : CAS PARTICULIER (merci d'être tolérant sur les valeurs en durs)
        // Ajustement pour l'affichage correcte des semaines (cas de décembre 2014 par exemple)
        // Si la dernière semaine d'une année ne comprend pas le jeudi dans cette année, il faut
        // recommencer à 1 le décompte
        if (mois == MOIS_DECEMBRE && jourMois >= 29)
            numSemaine = 1;

        cout << setw(ESPACE_SEMAINE) << numSemaine << " ";

        // SEMAINE : CAS PARTICULIER (merci d'être tolérant sur les valeurs en durs)
        // Si le nombre de semaine est trop grand, donc il est reporté depuis l'année
        // précédente car le premier jour de l'année courante est après le premier jeudi,
        // il faut réinitialiser la numérotation des semaines
        if (mois == MOIS_JANVIER && numSemaine >= NOMBRE_LIMITE_REPORT_SEMAINE)
            numSemaine = 0;
    }

    // On affiche d'abord des espaces pour commencer le mois le bon jour de la semaine
    cout << setw(ESPACE_NUMERO);
    if (compteur < debutDuMois)
        cout << " ";
    // Puis on affiche les dates
    else
        cout << jourMois++;

    // Si on a terminé une semaine :
    if (!(compteur % 7)) {
        // On incrémente numSemaine pour l'afficher la prochaine fois
        ++numSemaine;

        // Et on fait un retour à la ligne si on n'est pas à la fin du mois
        if (jourMois <= nbrJours)
            cout << endl;
    }
}

// Pour trouver le jour à retourner, on fait le décalage du mois précédent + le nombre de jour de
// ce mois et on prend le modulo pour trouver le jour de la semaine auquel il correspond. Le +1
// assure le décalage d'un jour
debutDuMois = ((nbrJours + debutDuMois - 1) % 7) + 1;
}

int saisieInt(const string messageSaisie, const int borneMin, const int borneMax) {
    int saisie;
    bool saisieOK;
    bool erreurFlux;

    const string MSG_ERREUR_FLUX = "Merci d'entrer un nombre entier.";
    const string MSG_ERREUR_VALEUR = "La valeur saisie n'est pas dans l'intervalle";

    do {
        cout << messageSaisie;
        erreurFlux = bool(cin >> saisie);
        saisieOK = false;

        if (!controleFlux(erreurFlux))
            cout << MSG_ERREUR_FLUX << endl;
        else if (saisie < borneMin || saisie > borneMax)
            cout << MSG_ERREUR_VALEUR << endl;
        else

```

```

    saisieOK = true;

    VIDER_BUFFER;
} while (!saisieOK);

return saisie;
}

bool saisieRecommencer(const char valeurVraieParam, const char valeurFausseParam) {
    // Pour préserver l'immutabilité de la fonction, nous demandons des paramètres constants
    // et les modifications dans des variables locales.

    // Définition des variables nécessaires à la saisie
    char saisie; // Variable qui contiendra la valeur saisie
    bool saisieOK,
        erreurFlux;

    const char VALEUR_VRAIE = (char) toupper(valeurVraieParam);
    const char VALEUR_FAUSSE = (char) toupper(valeurFausseParam);

    // Définition des constantes
    const string MSG_ERREUR_FLUX = "Veuillez entrer un caractere.";
    const string MSG_ERREUR_SAISIE = "La valeur saisie n'est pas une valeur possible.";

    do {
        // On récupère la saisie de l'utilisateur
        cout << "Voulez-vous recommencer [" << VALEUR_VRAIE << "/" << VALEUR_FAUSSE << "] ? ";
        erreurFlux = bool(cin >> saisie);
        saisieOK = false;

        // S'il y a eu une erreur de flux, on la corrige et on reboucle (flag saisieOK à false)
        if (!controleFlux(erreurFlux)) {
            cout << MSG_ERREUR_FLUX << endl;
            // Si la valeur n'est pas parmi les valeurs demandées, on reboucle
            else if (toupper(saisie) != VALEUR_VRAIE && toupper(saisie) != VALEUR_FAUSSE)
                cout << MSG_ERREUR_SAISIE << endl << endl;
            // Sinon, on accepte la valeur et on arrête de boucler (flag saisieOK à true)
            else
                saisieOK = true;

            VIDER_BUFFER;
        } while (!saisieOK);

        // On retourne finalement un booléen en fonction de la valeur saisie (true si 'o' ou 'O')
        return (toupper(saisie) == VALEUR_VRAIE);
    }

    bool verifieIntervalle(const int moisDebut, const int anneeDebut,
                           const int moisFin, const int anneeFin) {
        return (anneeDebut < anneeFin) || (anneeDebut == anneeFin && moisDebut <= moisFin);
    }

    int nbrJoursMois(const int mois, const int annee) {
        // On compte 31 pour tous les mois puis
        // - si c'est février, on déduit 2 si c'est bissextile et 3 si ça ne l'est pas
        // - si ce n'est pas février, on fait modulo 7 pour "mettre août sur janvier"
        // ensuite on enlève soit 0 soit 1 avec le modulo 2.

        return (31 - ((mois == 2) ? (3 - (int) estBissextile(annee)) : ((mois-1) % 7 % 2)));
    }

    bool estBissextile(const int anneeUtilisateur) {
        return bool(!(anneeUtilisateur % 400) || (!(anneeUtilisateur % 4) && (anneeUtilisateur % 100)));
    }

    void afficheBarre(char symbole, const char symboleEncadre, const int nbrSymbole) {
        //setfill n'accepte pas de const
        char resetFill = ' ';

        //affiche une suite d'un symbole encadré selon un nombre donné;
        cout << symboleEncadre << setw(nbrSymbole) << setfill(symbole) << symboleEncadre
            << endl;
        cout << setfill(resetFill);
    }

    char intJourEnChar(const int numJourSemaine) {
        char jour = ' ';

        // On match le jour de la semaine avec la première lettre du jour en char
        switch (numJourSemaine) {
            case 1:
                jour = 'L';
                break;
            case 2:
            case 3:
                jour = 'M';
                break;
            case 4:
                jour = 'J';
                break;
            case 5:
                jour = 'V';

```

```

        break;
    case 6:
        jour = 'S';
        break;
    case 7:
        jour = 'D';
        break;
    default:
        cout << "Le numero (" << numJourSemaine
              << ") ne correspond à aucun jour de la semaine." << endl;
    }

    return jour;
}

string intMoisEnString(const int numMois) {
    string mois = "";

    switch (numMois) {
        case 1:
            mois = "Janvier";
            break;
        case 2:
            mois = "Fevrier";
            break;
        case 3:
            mois = "Mars";
            break;
        case 4:
            mois = "Avril";
            break;
        case 5:
            mois = "Mai";
            break;
        case 6:
            mois = "Juin";
            break;
        case 7:
            mois = "Juillet";
            break;
        case 8:
            mois = "Aout";
            break;
        case 9:
            mois = "Septembre";
            break;
        case 10:
            mois = "Octobre";
            break;
        case 11:
            mois = "Novembre";
            break;
        case 12:
            mois = "Decembre";
            break;
        default:
            cout << "Le numero (" << numMois
                  << ") ne correspond a aucun mois de l'annee." << endl;
    }

    return mois;
}

bool controleFlux(const bool saisie) {
    bool saisieOK = true;

    if (!saisie) {
        cin.clear();
        saisieOK = false;
    }

    return saisieOK;
}

int jourSemaine(const int jour, const int mois, const int annee) {
    int d = jour,
        m = mois,
        y = annee;

    if (m <= 2) {
        m += 12;
        --y;
    }

    return (d + 2*m + 3*(m+1)/5 + y + (y/4) - (y/100) + (y/400) + 7) % 7 + 1;
}

int totalJourMois(const int moisFinal, const int annee) {
    const int MOIS_MIN = 1;
    int total = 0;

    for(int mois = MOIS_MIN; mois < moisFinal; ++mois)

```

```
        total += nbrJoursMois(mois, annee);

    return total;
}

int numeroSemainePourDate(const int jour, const int mois, const int annee) {
    const int NUM_JEUDI = 4;

    int joursMoisPassés = totalJourMois(mois, annee);
    int jeudiDeLaSemaine = jour + NUM_JEUDI - jourSemaine(jour, mois, annee);
    int joursTotaux = joursMoisPassés + jeudiDeLaSemaine;

    return (joursTotaux - 1)/7 + 1;
}

void ajusterNumeroPremiereSemaine(int &premiereSemaine, const int mois, const int annee) {
    const int CAS_SPECIAL_JAN_PREMIER_MOIS_AFFICHE = -1;
    const int NBR_JOUR_DECEMBRE = 31;
    const int MOIS_JANVIER = 1;
    const int MOIS_MAX = 12;

    // Si on commence une nouvelle année
    if (mois == MOIS_JANVIER) {
        // Puisque le premier jeudi de l'année est compris dans la première semaine,
        // si le premier jour de l'année est un jeudi ou avant, on recommence à un
        if (jourSemaine(1, mois, annee) <= 4) {
            premiereSemaine = 1;
        }
        // si le premier jour de l'année est un vendredi ou après, on est dans la dernière
        // semaine de l'année précédente.
        else if (premiereSemaine == CAS_SPECIAL_JAN_PREMIER_MOIS_AFFICHE) {
            premiereSemaine = numeroSemainePourDate(NBR_JOUR_DECEMBRE, MOIS_MAX, annee-1);
        }
    }
    // Si ce n'est pas janvier, on utilise une formule pour trouver la première semaine
    else {
        premiereSemaine = numeroSemainePourDate(1, mois, annee);
    }
}
```