

/\*

```

-----
Nom du fichier : main.cpp
Nom du labo : Labo 06 : crible Eratosthène
Auteur(s) : Laetitia Guidetti et Antonio Pollino
Date creation : 19.11.2021
Description : Ce programme permet d'identifier tous les nombres premiers,
              jusqu'à un nombre donné par la saisie de l'utilisateur. Pour
              cela, la méthode employée est celle du crible d'Eratosthène.

              Pour plus d'informations concernant la méthode d'Eratosthène :
              https://fr.wikipedia.org/wiki/Crible\_d%27%C3%89ratosth%C3%A8ne

Remarque(s) : La saisie de l'utilisateur est contrôlée, si une valeur ne
              correspond pas au spectre ou n'est pas un nombre entier, le
              programme demande l'entrée d'une nouvelle valeur.

Compilateur : Mingw-w64 g++ 11.1.0
-----

```

```

*/
#include <cstdlib>           // EXIT_SUCCESS
#include <iostream>         // cout
#include <limits>           // numeric_limits
#include <string>           // type string

#include "saisie.h"         // fonction saisie
#include "gestionTableau.h" // fonctions afficherTableau, positionValeur,
                           // afficherPositionValeur
#include "crible.h"         // fonction criblageTableau

using namespace std;

int main() {

    //-----
    // Constantes
    //-----
    // Messages
    const string MSG_SAISIE      = "Nombre de valeurs"s;
    const string MSG_ERREUR     = "Erreur de saisie"s;
    const string MSG_QUITTER    = "Presser ENTER pour quitter"s;
    const string MSG_INIT_TAB   = "Initialisation du tableau"s;
    const string MSG_CRIBLAGE   = "Criblage du tableau"s;
    const string MSG_NBR_PREMIERS = "Nombre de premiers : "s;

    // Constantes fonctionnelles
    const int MIN_SAISIE      = 2;
    const int MAX_SAISIE      = 100;
    const int DECALAGE        = 1;
    const char AFFICHAGE_VRAI = 'X';
    const char AFFICHAGE_FAUX = '0';
    const bool PREMIER        = false;
    const bool PAS_PREMIER    = true;

    //-----
    // Bienvenue
    //-----
    cout << "Ce programme identifie tous les nombres premiers jusqu'au nombre "
          "choisi par l'utilisateur, " << endl;
    cout << "selon la methode du crible d'Eratosthene" << endl << endl;

    //-----
    // Saisie et initialisation du tableau
    //-----
    const unsigned TAILLE = (unsigned)(DECALAGE +
                                       saisie(MIN_SAISIE, MAX_SAISIE, MSG_SAISIE, MSG_ERREUR));
    bool tableau[TAILLE];
    for(unsigned i = 0; i < TAILLE; ++i) {
        tableau[i] = false;
    }

    cout << endl;
    cout << MSG_INIT_TAB << endl;
    afficherTableau(tableau, DECALAGE, TAILLE - 1, AFFICHAGE_VRAI, AFFICHAGE_FAUX);
    cout << endl;

```

```
//-----  
// Identification des nombres premiers  
//-----  
unsigned position = 2;  
int      compteur = 0;  
  
tableau[1] = true; // 1 est un cas particulier, ni premier, ni composé  
  
// Crible d'Eratosthène  
while (position < TAILLE) {  
  
    criblageTableau(tableau, TAILLE, PAS_PREMIER, position);  
    ++position;  
    ++compteur;  
    position = positionValeur(tableau, TAILLE, PREMIER, position);  
}  
  
//-----  
// Affichage du criblage et des nombres premiers  
//-----  
cout << MSG_CRIBLAGE << endl;  
afficherTableau(tableau, DECALAGE, TAILLE - 1, AFFICHAGE_VRAI, AFFICHAGE_FAUX);  
cout << endl;  
  
cout << MSG_NBR_PREMIERS << compteur << endl;  
afficherPositionValeur(tableau, DECALAGE, TAILLE - 1, PREMIER);  
cout << endl;  
  
//-----  
// Fin de programme  
//-----  
cout << MSG_QUITTER;  
cin.ignore(numeric_limits<streamsize>::max(), '\n');  
return EXIT_SUCCESS;  
}
```

/\*

```
-----
Nom du fichier : saisie.cpp
Nom du labo    : Labo 06 : crible Eratosthène
Auteur(s)     : Laetitia Guidetti et Antonio Pollino
Date creation  : 19.11.2021
Description    : Déclaration de la fonction permettant d'obtenir une saisie
                  utilisateur contrôlée.
Remarque(s)   : -
Compilateur   : Mingw-w64 g++ 11.1.0
-----
```

\*/

```
#include <iostream>      // cout, cin
#include <string>         // type string
#include <limits>         // numeric_limits
#include "saisie.h"

using namespace std;

int saisie(int min, int max, const string& saisieMessage,
           const string& erreurMessage) {

    bool erreur;
    int  saisie;

    do {
        // message et saisie
        cout << saisieMessage << " [" << min << " - " << max << "]" :";

        // vérification
        erreur = !(cin >> saisie) || saisie < min || saisie > max;
        if (erreur) {
            cout << erreurMessage << endl;
            cin.clear();
        }

        // vider buffer
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    } while(erreur);
    return saisie;
}
```

/\*

```
-----
Nom du fichier : saisie.h
Nom du labo    : Labo 06 : crible Eratosthène
Auteur(s)     : Laetitia Guidetti et Antonio Pollino
Date creation  : 19.11.2021
Description    : Définition de la fonction permettant d'obtenir une saisie
                  utilisateur contrôlée entre une valeur minimale et une valeur
                  maximale.
Remarque(s)   : -
Compilateur    : Mingw-w64 g++ 11.1.0
-----
```

\*/

```
#ifndef LAB006_CRIBLE_SAISIE_H
#define LAB006_CRIBLE_SAISIE_H

#include <string>    // type string

/// Nom                saisie
/// But                Permet de récupérer une saisie d'utilisateur contrôlée
///
/// \param min          Valeur minimale comprise
/// \param max          Valeur maximal comprise
/// \param saisieMessage Message à afficher lors de la saisie
/// \param erreurMessage Message à afficher lors d'une erreur de saisie
/// \return             Retourne la valeur saisie de l'utilisateur en int
int saisie(int min, int max, const std::string& saisieMessage,
           const std::string& erreurMessage);

#endif //LAB006_CRIBLE_SAISIE_H
```

```
/*
-----
Nom du fichier : gestionTableau.cpp
Nom du labo    : Labo 06 : crible Eratosthène
Auteur(s)     : Laetitia Guidetti et Antonio Pollino
Date creation  : 19.11.2021
Description    : Définition des fonctions permettant l'affichage, et la
                  récupération d'informations d'un tableau.
Remarque(s)   : -
Compilateur   : Mingw-w64 g++ 11.1.0
-----
*/

#include <iostream>          // cout
#include <iomanip>           // setw

#include "gestionTableau.h"

using namespace std;

void afficherTableau(const bool tab[], unsigned min, unsigned max,
                    char vrai, char faux, unsigned colonne) {

    unsigned nombreColonnes = 0;
    for(unsigned position = min; position <= max; ++position) {
        tab[position] ? cout << vrai : cout << faux;
        cout << " ";

        // Retour à la ligne lorsque colonne est atteint
        if(++nombreColonnes == colonne && position != max) {
            cout << endl;
            nombreColonnes = 0;
        }
    }
    cout << endl;
}

unsigned positionValeur(const bool tab[], unsigned taille, bool valeur,
                       unsigned position) {

    for(; position < taille; ++position) {
        if (tab[position] == valeur){
            return position;
        }
    }
    return taille;
}

void afficherPositionValeur(const bool tab[], unsigned min, unsigned max,
                           bool valeur, unsigned colonne, int alignement) {

    unsigned nombreColonnes = 0;
    for(unsigned position = min; position <= max; ++position) {
        if (valeur == tab[position]) {
            cout << setw(alignement) << position << " ";

            // Retour à la ligne lorsque colonne est atteint
            if(++nombreColonnes == colonne && position != max) {
                cout << endl;
                nombreColonnes = 0;
            }
        }
    }
    cout << endl;
}
```

/\*

```

-----
Nom du fichier : gestionTableau.h
Nom du labo    : Labo 06 : crible Eratosthène
Auteur(s)     : Laetitia Guidetti et Antonio Pollino
Date creation  : 19.11.2021
Description    : Déclaration des fonctions permettant l'affichage d'un segment
                  précis ou d'une position d'un tableau, ainsi que la récupération
                  d'informations telle que la position d'une valeur dans un tableau.
Remarque(s)   : -
Compilateur    : Mingw-w64 g++ 11.1.0
-----

```

\*/

```

#ifndef LABO06_CRIBLE_GESTIONTABLEAU_H
#define LABO06_CRIBLE_GESTIONTABLEAU_H

```

```

/// Nom          afficherTableau
/// But          Afficher un tableau depuis un index minimal jusqu'à un index
///              maximal. Le caractère affiché en fonction de la valeur du
///              tableau est à choisir.
///

```

```

/// \param tab    Le tableau à afficher
/// \param min     L'index minimal depuis lequel l'affichage commence
/// \param max     L'index maximal où l'affichage se finit. Il doit être
///              strictement plus petit que la taille du tableau
/// \param vrai    Caractère à afficher quand la valeur est égale à 1
/// \param faux    Caractère à afficher quand la valeur est égale à 0
/// \param colonne Nombre de colonnes de l'affichage

```

```

void afficherTableau(const bool tab[], unsigned min, unsigned max,
                    char vrai, char faux, unsigned colonne = 10);

```

```

/// Nom          positionValeur
/// But          Retourner la position du premier élément trouvé
///              correspondant à la valeur passé en paramètre
///

```

```

/// \param tab    Le tableau à afficher
/// \param taille  Taille du tableau
/// \param valeur  Valeur recherchée
/// \param position Position depuis laquelle il faut chercher la valeur
/// \return        Retourne la position du premier élément trouvé

```

```

unsigned positionValeur(const bool tab[], unsigned taille, bool valeur,
                      unsigned position);

```

```

/// Nom          afficherPositionValeur
/// But          Afficher les positions correspondantes à la valeur entrée
///              en paramètre
///

```

```

/// \param tab    Le tableau à afficher
/// \param min     L'index minimal depuis lequel l'affichage commence
/// \param max     L'index maximal où l'affichage se finit. Il doit être
///              strictement plus petit que la taille du tableau
/// \param valeur  Valeur dont les positions doivent être affichées
/// \param colonne Nombre de colonnes de l'affichage
/// \param alignement Espacement d'un nombre à l'autre lors de l'affichage

```

```

void afficherPositionValeur(const bool tab[], unsigned min, unsigned max,
                          bool valeur, unsigned colonne = 10, int alignement = 2);

```

```

#endif //LABO06_CRIBLE_GESTIONTABLEAU_H

```

```
/*
-----
Nom du fichier : crible.cpp
Nom du labo    : Labo 06 : crible Eratosthène
Auteur(s)      : Laetitia Guidetti et Antonio Pollino
Date creation  : 20.11.2021
Description    : Définition de la fonction permettant de cribler le tableau.
Remarque(s)    : -
Compilateur    : Mingw-w64 g++ 11.1.0
-----
*/

#include "crible.h"

void criblageTableau(bool tab[], unsigned taille, bool valeur,
                    unsigned diviseur) {

    unsigned deplacement = diviseur;
    diviseur *= diviseur;

    for(; diviseur < taille; diviseur += deplacement) {
        tab[diviseur] = valeur;
    }
}
```

```
/*
-----
Nom du fichier : crible.h
Nom du labo : Labo 06 : crible Eratosthène
Auteur(s) : Laetitia Guidetti et Antonio Pollino
Date creation : 20.11.2021
Description : Déclaration de la fonction permettant de cribler le tableau en
fonction de la taille et du caractère souhaité.
Remarque(s) : -
Compilateur : Mingw-w64 g++ 11.1.0
-----
*/

#ifndef LABO06_CRIBLE_CRIBLE_H
#define LABO06_CRIBLE_CRIBLE_H

/// Nom cribleTableau
/// But Remplacer les valeurs dont la position est divisible par
/// le diviseur et est plus grande ou égale à diviseur^2
///
/// \param tab Le tableau à modifier
/// \param taille La taille du tableau
/// \param valeur La nouvelle valeur à insérer dans le tableau
/// \param diviseur Valeur utilisée pour tester si la position est un multiple
/// de diviseur et est plus grande ou égale à diviseur^2
void cribleTableau(bool tab[], unsigned taille, bool valeur,
unsigned diviseur);

#endif //LABO06_CRIBLE_CRIBLE_H
```