

```

1  /*
2  -----
3  Nom du fichier : main.cpp
4  Nom du Labo : Labo 6 - Cribles
5
6  Auteur(s) : Grégoire Guyot et Théo Coutaudier
7
8  Date creation : 19-11-2021
9
10 Description : L'utilisateur choisi une valeur entre 2 et 100, le programme va
11 créer un tableau avec le nombre de valeur choisis par l'utilisateur,
12 un algorithme va chercher les nombres premier entre 0 et la
13 valeur saisie.
14
15 Remarque(s) : La saisie de l'utilisateur est contrôlée
16 Le constante peuvent être modifier pour afficher les tableau sur
17 plus/moins de colonne ou plus d'espace entre les valeurs
18 Lors de la création du tableau, il y a un avertissement, car le
19 tableau pourrait être créée, mais cela n'arrive pas.
20
21 Compilateur : Mingw-w64 g++ 11.2.0
22 -----
23 */
24
25 #include <cstdlib>
26 #include <iostream>
27 #include <limits> // utilisée pour vider le buffer avec numeric_limits<streamsize>
28 #include "affichageTableau.h"
29 #include "countListTableau.h"
30 #include "saisieVerification.h"
31 #include "initTableau.h"
32 #include "checkNbPremier.h"
33
34 using namespace std;
35
36 int main() {
37
38     //-----
39     // Déclaration constante
40     //-----
41
42     const unsigned int MIN = 2; //valeur minimum testSaisie
43     const unsigned int MAX = 100; //valeur maximum testSaisie
44     const int space = 3; //espace pour le setw()
45     const int nbcolonne = 10; //définir taille affichage tableau
46     const bool valeur = true; //initTableau
47
48
49     //-----
50     // Saisie utilisateur et création tableau
51     //-----
52
53     cout << "Nombre de valeurs [" << MIN << " - " << MAX << "] : ";
54     //message d'avertissement volontairement ignoré pour size_t
55     //car la valeur va être réutilisé dans le futur en size_t
56     size_t taille = size_t(testSaisie(MIN, MAX));
57     bool tab[taille];
58
59     //-----
60     // Initialisation tableau et affichage
61     //-----
62
63     cout << endl << "initialisation du tableau... " << endl;
64     initTableau(tab, taille, valeur); //initialisation du tableau
65     afficherCribles(tab, taille, nbcolonne, space); //affichage du tableau
66
67     //-----
68     // Modification tableau et affichage
69     //-----
70
71     cout << endl << "modification du tableau... " << endl;
72     criblage(tab, taille); //modification du tableau
73     afficherCribles(tab, taille, nbcolonne, space); //affichage du tableau
74
75     //-----
76     // Affichage liste des nombres premiers
77     //-----

```

```

78
79     cout << endl << "Il y a " << countTrueElement(tab, taille)
80         << " nombre premier" << endl;
81     //liste de tout les nombres premiers
82     listBoolElement(tab, taille, nbcolonne , space, valeur);
83
84     //-----
85     // Fin de programme
86     //-----
87
88     cout << endl << "Merci utilise le programme. Appuyez sur ENTER pour terminer le "
89         "programme...";
90     cin.ignore(numeric_limits<streamsize>::max(), '\n');
91     return EXIT_SUCCESS;
92 }
93
94 /*
95 -----
96 Nom du fichier :   initTableau.h
97 Nom du labo      :   Labo 6 - Cribles
98
99 Auteur(s)       :   Théo Coutaudier et Grégoire Guyot
100
101 Date creation   :   19-11-2021
102
103 Description     :   Ajoute une valeur par défaut les cases du tableau
104
105 Remarque(s)    :   Ne peut remplir que les valeurs par défaut true ou false
106
107 Compilateur    :   Mingw-w64 g++ 11.1.0
108 -----
109 */
110
111 #ifndef LABOCRIBLES_INITTABLEAU_H
112 #define LABOCRIBLES_INITTABLEAU_H
113
114 ///
115 /// Nom : initTableau
116 /// But : Ajoute une valeur par défaut à toute les cases d'un tableau booléen
117 ///
118 /// \param tab      le tableau à initialiser (tableau bool)
119 /// \param taille    la taille du tableau (en size_t)
120 /// \param valeur    la valeur bool avec la quelle initialiser le tableau (true false)
121 /// Exception : n/a
122 void initTableau(bool tab[], size_t taille, bool valeur);
123
124 #endif //LABOCRIBLES_INITTABLEAU_H
125
126 /*
127 -----
128 Nom du fichier :   initTableau.cpp
129 Nom du labo      :   Labo 6 - Cribles
130
131 Auteur(s)       :   Théo Coutaudier et Grégoire Guyot
132
133 Date creation   :   19-11-2021
134
135 Description     :   Ajoute une valeur par défaut les cases du tableau
136
137 Remarque(s)    :   Ne peut remplir que les valeurs par défaut true ou false
138
139 Compilateur    :   Mingw-w64 g++ 11.1.0
140 -----
141 */
142
143 #include <cstdlib>
144 #include "initTableau.h"
145
146 using namespace std;
147
148 void initTableau(bool tab[], size_t taille, bool valeur) {
149
150     //initalise chaque case du tableau à valeur
151     for (size_t i = 0; i < taille; ++i) {
152         tab[i] = valeur;
153     }
154 }

```

```

155
156  /*
157  -----
158  Nom du fichier :  countListTableau.cpp
159  Nom du Labo    :  Labo 6 - Cribles
160
161  Auteur(s)      :  Grégoire Guyot et Théo Coutaudier
162
163  Date creation   :  19-11-2021
164
165  Description     :  La fonction countTrueElement compte le nombre de valeur à true
166                    dans le tableau.
167                    La fonction listTrueElement liste les cases du tableau qui sont
168                    à true.
169
170  Remarque(s)    :  n/a
171
172  Compilateur     :  Mingw-w64 g++ 11.2.0
173  -----
174  */
175
176  #ifndef LABOCRIBLES_COUNTLISTTABLEAU_H
177  #define LABOCRIBLES_COUNTLISTTABLEAU_H
178  ///
179  /// Nom :  countTrueElement
180  /// But :  Compte le nombre de valeur True dans un tableau
181  ///
182  /// \param tab Le tableau à analyser
183  /// \param taille La taille du tableau en size_t
184  /// \return Le nombre de valeur True dans le tableau en size_t
185  /// Exception : n/a
186  size_t countTrueElement(const bool tab[], size_t taille);
187
188  ///
189  /// Nom :  listBoolElement
190  /// But :  Affiche la position de chaque élément à true ou false
191  ///
192  /// \param tab          tableau (bool)
193  /// \param taille       taille du tableau (size_t)
194  /// \param nbParLigne   Nombre d'élément par ligne
195  /// \param espace       espace d'affichage entre chaque élément (int)
196  /// \param valeur       bool indiquant s'il faut récupérer les valeurs true ou false
197  /// Exception : n/a
198  void listBoolElement(const bool tab[], size_t taille, int nbcolonne, int espace,
199                      bool valeur);
200
201  #endif //LABOCRIBLES_COUNTLISTTABLEAU_H
202
203  /*
204  -----
205  Nom du fichier :  countListTableau.cpp
206  Nom du Labo    :  Labo 6 - Cribles
207
208  Auteur(s)      :  Grégoire Guyot et Théo Coutaudier
209
210  Date creation   :  19-11-2021
211
212  Description     :  La fonction countTrueElement compte le nombre de valeur à true
213                    dans le tableau.
214                    La fonction listTrueElement liste les cases du tableau qui sont
215                    à true.
216
217  Remarque(s)    :  n/a
218
219  Compilateur     :  Mingw-w64 g++ 11.2.0
220  -----
221  */
222
223  #include <limits>
224  #include <cstdlib>
225  #include <iostream>
226  #include <iomanip>
227  #include "countListTableau.h"
228  using namespace std;
229
230  size_t countTrueElement(const bool tab[], size_t taille){
231      size_t count = 0; //initialisation du counter qui sera retourner

```

```

232     for (size_t i = 0; i < taille; ++i) {
233         if (tab[i]) {
234             count++;
235         }
236     }
237     return count;
238 }
239 void listBoolElement(const bool tab[], size_t taille, int nbParLigne, int espace,
240                     bool valeur) {
241     int count = 0;
242     for (size_t i = 1; i < taille; ++i) {
243         if(tab[i]==valeur) {
244             cout << setw(espace) << i + 1;
245             count++;
246             //vérifie si le programme doit passé à la ligne
247             if (!(count%nbParLigne) && count!=0){
248                 cout<<endl;
249             }
250         }
251     }
252     cout << endl;
253 }
254
255 /*
256 -----
257 Nom du fichier : affichageTableau.h
258 Nom du Labo    : Labo 6 - Cribles
259
260 Auteur(s)      : Grégoire Guyot et Théo Coutaudier
261
262 Date creation  : 19-11-2021
263
264 Description    : Affiche un tableau de bool en affichant un 'O' lorsque la valeur
265                  est true et 'X' lorsque la valeur est false.
266
267 Remarque(s)   : n/a
268
269 Compilateur    : Mingw-w64 g++ 11.2.0
270 -----
271 */
272
273 #ifndef LABOCRIBLES_AFFICHAGETABLEAU_H
274 #define LABOCRIBLES_AFFICHAGETABLEAU_H
275
276
277 /// Nom : afficherCribles
278 /// But : afficher le tableau
279 ///
280 /// \param tab tableau à afficher
281 /// \param taille taille du tableau
282 /// \param nbColonnes nombre de colonnes à afficher
283 /// \param space nombre d'espaces entre les caractères du tableau
284 /// Exceptions : n/a
285 void afficherCribles(const bool tab[], size_t taille, int nbColonnes, int space);
286
287 #endif //LABOCRIBLES_AFFICHAGETABLEAU_H
288
289 /*
290 -----
291 Nom du fichier : affichageTableau.cpp
292 Nom du Labo    : Labo 6 - Cribles
293
294 Auteur(s)      : Grégoire Guyot et Théo Coutaudier
295
296 Date creation  : 19-11-2021
297
298 Description    : Affiche un tableau de bool en affichant un 'O' lorsque la valeur
299                  est true et 'X' lorsque la valeur est false.
300
301 Remarque(s)   : n/a
302
303 Compilateur    : Mingw-w64 g++ 11.2.0
304 -----
305 */
306
307 #include <cstdlib>
308 #include <iostream>

```

```

309 #include <iomanip>
310 #include "affichageTableau.h"
311
312 using namespace std;
313
314 void afficherCribles(const bool tab[], size_t taille, int nbColonnes, int space){
315     for (size_t i = 0; i < taille; ++i){ // int ou size_t
316         if (!(i % (static_cast<unsigned long long int>(nbColonnes))) && i != 0){
317             cout << endl;
318         }
319         cout << setw(space) << (tab[i] ? '0' : 'X');
320     }
321     cout << endl;
322 }
323
324 /*
325 -----
326 Nom du fichier : checkNbPremier.h
327 Nom du Labo    : Labo 6 - Cribles
328
329 Auteur(s)      : Grégoire Guyot et Théo Coutaudier
330
331 Date creation  : 19-11-2021
332
333 Description     : Utilise le théorème de Erathostène pour cribler le tableau et
334                  mettre à false les nombre qui ne sont pas premier. Commence par 2
335                  puis trouve le prochain diviseur premier donc 3 puis crible le
336                  tableau à nouveau.
337
338 Remarque(s)    : Le premier diviseur est initialisé à 2 car c'est le premier nombre
339                  premier, c'est le cas dans le criblage d'Eratosthène également.
340
341 Compilateur     : Mingw-w64 g++ 11.2.0
342 -----
343 */
344
345 #ifndef LABOCRIBLES_CHECKNBPREMIER_H
346 #define LABOCRIBLES_CHECKNBPREMIER_H
347
348
349 /// Nom : criblage
350 /// But : crible le tableau en commençant par la valeur 2
351 ///
352 /// \param tab le tableau à cribler
353 /// \param taille la taille du tableau à cribler
354 /// Exception : n/a
355 void criblage(bool tab[], size_t taille);
356
357 /// Nom : criblageDiv
358 /// But : utilise le diviseur actuel et crible le tableau avec en mettant à false
359 ///       : les nombres qui sont divisibles par le diviseur actuel
360 ///
361 /// \param tab le tableau à cribler
362 /// Exception : n/a
363 void criblageDiv(bool tab[]);
364
365 /// Nom : prochainDiv
366 /// But : trouver le prochain diviseur avec lequel cribler le tableau
367 ///
368 /// \param tab à cribler
369 /// \return le prochain diviseur de size_t pour éviter les conversions avec les
370 ///         autres types lorsque il est utilisé dans les calculs
371 /// Exception : n/a
372 size_t prochainDiv(const bool tab[]);
373 #endif //LABOCRIBLES_CHECKNBPREMIER_H
374
375 /*
376 -----
377 Nom du fichier : checkNbPremier.cpp
378 Nom du Labo    : Labo 6 - Cribles
379
380 Auteur(s)      : Grégoire Guyot et Théo Coutaudier
381
382 Date creation  : 19-11-2021
383
384 Description     : Utilise le théorème de Erathostène pour cribler le tableau et
385                  mettre à false les nombre qui ne sont pas premier. Commence par 2

```

```

386                puis trouve le prochain diviseur premier donc 3 puis crible le
387                tableau à nouveau.
388
389 Remarque(s)      : Le premier diviseur est initialisé à 2 car c'est le premier nombre
390                    premier, c'est le cas dans le criblage d'Eratosthène également.
391
392 Compilateur      : Mingw-w64 g++ 11.2.0
393 -----
394 */
395
396 #include <cstdlib>
397 #include "checkNbPremier.h"
398
399 using namespace std;
400
401 size_t tailleTableau;
402 size_t diviseur = 2; //Commence à 2
403
404 void criblage(bool tab[], size_t taille){
405     tailleTableau = taille;
406     //Exception de tab[0] qui n'est pas prise en compte par criblageDiv
407     tab[0] = false;
408     while(diviseur * diviseur < tailleTableau){
409         criblageDiv(tab);
410         diviseur = prochainDiv(tab);
411     }
412
413 }
414
415 void criblageDiv(bool tab[]){
416     for(size_t i = diviseur; i <= tailleTableau; ++i){
417         if (i % diviseur == 0 && i != diviseur){
418             tab[i-1] = false;
419         }
420     }
421 }
422
423 size_t prochainDiv(const bool tab[]){
424     for(size_t i = diviseur; i < tailleTableau; ++i){
425         if(tab[i]){
426             return ++i;
427         }
428     }
429 }
430 // Le warning est présent car le return est englobé dans une condition qui se
431 // trouve dans une boucle, le compilateur propose d'ajouter un return 0; bien qu'on
432 // entre toujours dans la boucle for qui contient le return.
433
434 /*
435 -----
436 Nom du fichier : saisieVerification.h
437 Nom du labo    : Labo 6 - Cribles
438
439 Auteur(s)     : Grégoire Guyot et Théo Coutaudier
440
441 Date creation  : 19-11-2021
442
443 Description    : Vérifie que la saisie de l'utilisateur est valide et conforme à la
444                  saisie demandés
445
446 Remarque(s)    : - Vérifie si la valeur est du bon type
447                  - Vérifie si la valeur est dans les bonnes bornes
448                  - Si la valeur est fausse alors un message d'erreur s'affiche
449                    et l'utilisateur doit resaisir une valeur
450
451 Compilateur    : Mingw-w64 g++ 11.2.0
452 -----
453 */
454
455 #ifndef LAB05_1_SAISIEVERIFICATION_H
456 #define LAB05_1_SAISIEVERIFICATION_H
457
458 ///
459 /// Nom : testSaisie
460 /// But : Demande à l'utilisateur de saisir un nombre
461 ///      et vérifie qu'elle correspond aux parametres ci-dessous
462 ///

```

```
463 /// \param saisieMin la valeur minimum que l'utilisateur peut saisir
464 /// \param saisieMax la valeur maximum que l'utilisateur peut saisir
465 /// \return la valeur saisie par l'utilisateur (en int)
466 /// Exception : n/a
467 int testSaisie(int saisieMin, int saisieMax);
468
469 #endif //LAB05_1_SAISIEVERIFICATION_H
470
471 /*
472 -----
473 Nom du fichier : saisieVerification.cpp
474
475 Nom du labo      : Labo 6 - Cribles
476
477 Auteur(s)       : Grégoire Guyot et Théo Coutaudier
478
479 Date creation   : 19-11-2021
480
481 Description     : Vérifie que la saisie de l'utilisateur est valide et conforme à la
482                  saisie demandés
483
484 Remarque(s)    : - Vérifie si la valeur est du bon type
485                  - Vérifie si la valeur est dans les bonnes bornes
486                  - Si la valeur est fausse alors un message d'erreur s'affiche
487                  et l'utilisateur doit resaisir une valeur
488
489 Compilateur    : Mingw-w64 g++ 11.2.0
490 -----
491 */
492
493 #include <iostream>
494 #include <limits>
495 #include <string>
496 #define VIDER_BUFFER cin.ignore(numeric_limits<streamsize>::max(), '\n')
497
498 using namespace std;
499
500 int testSaisie(int saisieMin, int saisieMax)
501 {
502     int nbreSaisies;
503     bool erreurSaisie;
504     do {
505         cin >> nbreSaisies;
506         //verifie que la saisie correspond au type
507         //verifie aussi que la saisie est entre saisieMin et saisieMax
508         erreurSaisie = (cin.fail() or
509                         (nbreSaisies < saisieMin or nbreSaisies > saisieMax));
510         if(erreurSaisie){
511             cin.clear();
512             cout << "Erreur saisie, valeurs autorisees ["
513                  << saisieMin << "-"<< saisieMax << "]" << endl;
514         }
515         VIDER_BUFFER;
516     } while(erreurSaisie);
517     return nbreSaisies;
518 }
519
520
```