Programmation avancée en C++ **GIF-1003**
H2022

Nom	Prénom	Matricule	Programme

Travail Pratique 1
Rapport d'investigation:

Cahier des charges

En observant le code, on peut noter que le programme tente de dresser un tableau qui permet de nous donner les notes d'une classe d'étudiants. En outre, le tableau nous donne plus d'informations (soit ici l'écart-type calculé à partir des notes des élèves ainsi que la note maximale et la note minimale de chaque valeur entrée aux examens). Cependant, le code présente une multitude d'erreurs qu'on devra corriger, on devra par conséquent employer une stratégie afin de produire un rapport complet de toutes les erreurs et tenter de les corriger.

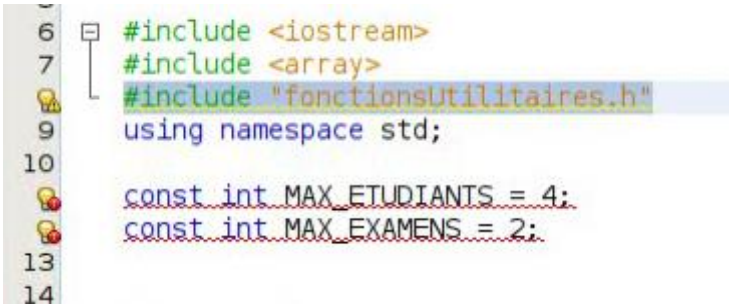
Stratégie

Notre stratégie repose sur plusieurs parties; la première c'est l'identification des erreurs de syntaxe (vu que le compilateur les détecte lui-même et nous explique pourquoi ces erreurs aboutissent) on procédera à leurs corrections, et de ce fait on en proposera des solutions. Une seconde partie se focalisera sur la détermination des erreurs de débordements et d'éditions de liens (ceux-là sont bien plus durs à identifier, c'est pour ceci qu'on devra faire appel au fameux débogueur). Une fois ceci résolu, et que le compilateur se compile parfaitement sans afficher aucune erreur sur la console, on passera à la partie suivante de notre stratégie qui sera d'exécuter le code afin de comparer notre résultat par rapport au résultat désiré (affichage désiré). Cette partie de notre stratégie va se

concentrer sur la détection des erreurs de logiques qu'on devra corriger pour produire un code correct et fonctionnel.

Erreur 1 : fichier 'programmePrincipal.cpp' ligne 11, 12

Localisation :



Justification:

C'est une erreur de syntaxe, les variables MAX_EXAMENS et MAX_ETUDIANTS sont déjà définies dans le fichier « fonctionsUtilitaires.h ».

Explications :

Les deux variables ont été définis dans le fichier fonctionsUtilitaires.h, et sont redéfinis dans le fichier « programmePrincipal.cpp ».

```
../investigationErreurs/programmePrincipal.cpp:11:11: error: redefinition of 'const int MAX_ETUDIANTS'
   11 | const int MAX_ETUDIANTS = 4;
      |           ^~~~~~~~~~~~~
In file included from ../investigationErreurs/programmePrincipal.cpp:8:
../investigationErreurs/fonctionsUtilitaires.h:14:11: note: 'const int MAX_ETUDIANTS' previously defined here
   14 | const int MAX_ETUDIANTS = 4;
      |           ^~~~~~~~~~~~~
../investigationErreurs/programmePrincipal.cpp:12:11: error: redefinition of 'const int MAX_EXAMENS'
   12 | const int MAX_EXAMENS = 2;
      |           ^~~~~~~~~~~
In file included from ../investigationErreurs/programmePrincipal.cpp:8:
../investigationErreurs/fonctionsUtilitaires.h:13:11: note: 'const int MAX_EXAMENS' previously defined here
   13 | const int MAX_EXAMENS = 2;
```

Solution :

On supprime les variables qui figurent dans le programme principal, et on inclut le fichier fonctionsUtilitaires.h dans le fichier « programmePrincipal.cpp ».

Explication :

Pour éviter la redondance, on inclut dans le code du programme principal tout simplement le fichier ou sont nommées toutes les variables; ici en l'occurrence « fonctionsUtilitaires.h ».

```
1  /**
2   * \file programmePrincipal.cpp
3   * \brief
4   */
5
6  #include <iostream>
7  #include <array>
   #include "fonctionsUtilitaires.h"
9  using namespace std;
10
11
12
13
14 int main ()
15 {
```

Erreur 2 : fichier programmePrincipal.cpp ligne 37.

Localisation

```
../../investigationErreurs2/programmePrincipal.cpp: In function 'int main()':
../../investigationErreurs2/programmePrincipal.cpp:37:3: error: 'afficherTableau' was not declared in this scope
   37 |   afficherTableau (tabNotes, &nbEleves);
      |   ^~~~~~~~~~~~~~
make[2]: *** [nbproject/Makefile-Debug.mk:74 : build/Debug/GNU-Linux/_ext/55f2309e/programmePrincipal.o] Erreur 1
make[2] : on quitte le répertoire « /mnt/hgfs/Partage/TP/ProjetInvestigationErreur »
make[1]: *** [nbproject/Makefile-Debug.mk:60 : .build-conf] Erreur 2
make[1] : on quitte le répertoire « /mnt/hgfs/Partage/TP/ProjetInvestigationErreur »
make: *** [nbproject/Makefile-impl.mk:40 : .build-impl] Erreur 2
```

Justification

C'est une erreur de mise en garde (soit une erreur de syntaxe), la fonction afficherTableau ne figure pas dans le fichier fonctionsUtilitaires.h; là ou toutes les fonctions sont censés être déclarées.

Explication

La fonction afficherTableau devrait faire partie des fonctions déclarées dans le programme du fichier fonctionsUtilitaires.h afin que le compilateur puisse la reconnaître.

Solution

On déclare la fonction afficherTableau dans le fichier fonctionsUtilitaires.h afin qu'il puisse la reconnaître.

```
void saisieNotes (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS > p_tabNotes, int*);
int minimum (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int*);
int maximum (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int*);
double ecartType (std::array<int, MAX_EXAMENS>& p_tabNotesEtudiant);
void afficherTableau (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves);
```

Erreur 3 : fichier fonctionsUtilitaires.cpp ligne 43.

Localisation :

```
g++-10    -c -g -MMD -MP -MF "build/Debug/GNU-Linux/_ext/55f2309e/fonctionsUtilitaires.o.d" -o build/Debug/GNU-Linux/_ext/55f2309e/fonctionsUtilitaires.o ../../investigationErreurs2/fonctionsUtilitaires.cpp
../../investigationErreurs2/fonctionsUtilitaires.cpp: In function 'int minimum(std::array<std::array<int, 2>, 4>&, int*)':
../../investigationErreurs2/fonctionsUtilitaires.cpp:43:1: warning: no return statement in function returning non-void [-Wreturn-type]
   43 | }
      | ^
mkdir -p build/Debug/GNU-Linux/_ext/55f2309e
rm -f "build/Debug/GNU-Linux/_ext/55f2309e/programmePrincipal.o.d"
g++-10    -c -g -MMD -MP -MF "build/Debug/GNU-Linux/_ext/55f2309e/programmePrincipal.o.d" -o build/Debug/GNU-Linux/_ext/55f2309e/programmePrincipal.o ../../investigationErreurs2/programmePrincipal.cpp
mkdir -p dist/Debug/GNU-Linux
```

Justification :

Ici, il n'y a pas de return vers la fin de la fonction. C'est une erreur de syntaxe.

Explication :

La fonction minimum est censée retourner la valeur de la noteFaible.

Solution :

On propose donc, de retourner la valeur de la noteFaible en l'ajoutant vers la fin du bloc de la fonction.

```
30  int
31  minimum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
32  {
33    int noteFaible = 100;
34    for (int i = 0; i < MAX_ETUDIANTS; i++)
35      {
36        for (int j = 0; j < MAX_EXAMENS; i++)
37          {
38            if (p_tabNotes[i][j] < noteFaible)
39              {
40                noteFaible = p_tabNotes[i][j];
41              }
42          }
43      }
44
45    return noteFaible;
46
    }
```

Erreur 4 : fichier programmePrincipal.cpp ligne 31

Localisation :

```
mkdir -p dist/Debug/GNU-Linux
g++-10     -o dist/Debug/GNU-Linux/projetinvestigationerreur build/Debug/GNU-Linux/_ext/55f2309e/fonctionsUtilitaires.o build/Debug/GNU-Linux/_ext/55f2309e/programmePrincipal.o
/usr/bin/ld : build/Debug/GNU-Linux/_ext/55f2309e/programmePrincipal.o : dans la fonction « main » :
/mnt/hgfs/Partage/TP/ProjetInvestigationErreur/../../investigationErreurs2/programmePrincipal.cpp:31 : référence indéfinie vers « saisieNotes(std::array<std::array<int, 2ul>, 4ul>, int*) »
collect2: error: ld returned 1 exit status
make[2]: *** [nbproject/Makefile-Debug.mk:64 : dist/Debug/GNU-Linux/projetinvestigationerreur] Erreur 1
make[2] : on quitte le répertoire « /mnt/hgfs/Partage/TP/ProjetInvestigationErreur »
make[1]: *** [nbproject/Makefile-Debug.mk:60 : .build-conf] Erreur 2
make[1] : on quitte le répertoire « /mnt/hgfs/Partage/TP/ProjetInvestigationErreur »
make: *** [nbproject/Makefile-impl.mk:40 : .build-impl] Erreur 2

BUILD FAILED (exit value 2, total time: 775ms)
```

On a une référence indéfinie vers la fonction saisieNotes qui se trouve dans le programme principal. C'est une erreur de syntaxe.

Justification :

Le nombre d'élèves devrait être un argument de la fonction saisieNotes.

Explication :

On va devoir ajouter le nombre d'élèves p_nbEleves comme argument de la fonction saisieNotes; en déclarant que c'est un nombre entier donc « int p_nbEleves ».

Solution :

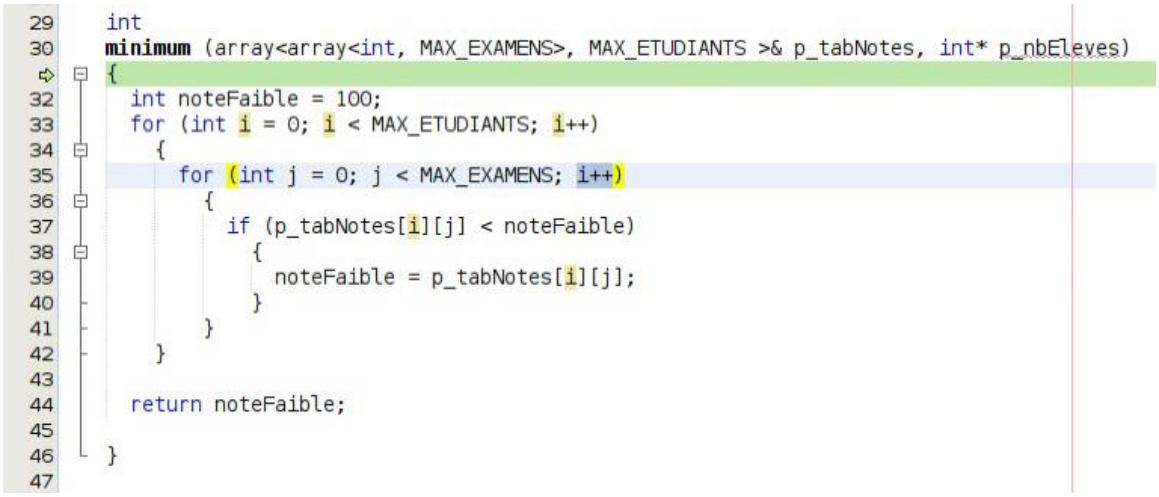
```
12
13    void
14    saisieNotes (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS > p_tabNotes, int* p_nbEleves)
15    {
16
17      for (int i = 0; i < *p_nbEleves; i++)
18        {
19          for (int j = 0; j < MAX_EXAMENS; j++)
20            {
21              cout << "Saisissez la note de l'examen " << j + 1 << endl;
22              cin >> p_tabNotes[i][j];
23            }
24        }
25    }
26
```

En prenant compte de la déclaration de la fonction saisieNotes qui a déjà été déclarée dans le fichier fonctionsUtilitaires.h en ajoutant int*.

```
16  void saisieNotes (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS > p_tabNotes, int*);
17  int minimum (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int*);
18  int maximum (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int*);
19  double ecartType (std::array<int, MAX_EXAMENS>& p_tabNotesEtudiant);
20  void afficherTableau (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves);
21
```

Erreur 5 : fichier fonctionsUtilitaires.cpp ligne 31

Localisation :



```
29  int
30  minimum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
31  {
32    int noteFaible = 100;
33    for (int i = 0; i < MAX_ETUDIANTS; i++)
34      {
35        for (int j = 0; j < MAX_EXAMENS; i++)
36          {
37            if (p_tabNotes[i][j] < noteFaible)
38              {
39                noteFaible = p_tabNotes[i][j];
40              }
41          }
42      }
43
44    return noteFaible;
45
46  }
47
```

On a un problème de segmentation, c'est une erreur de logique.

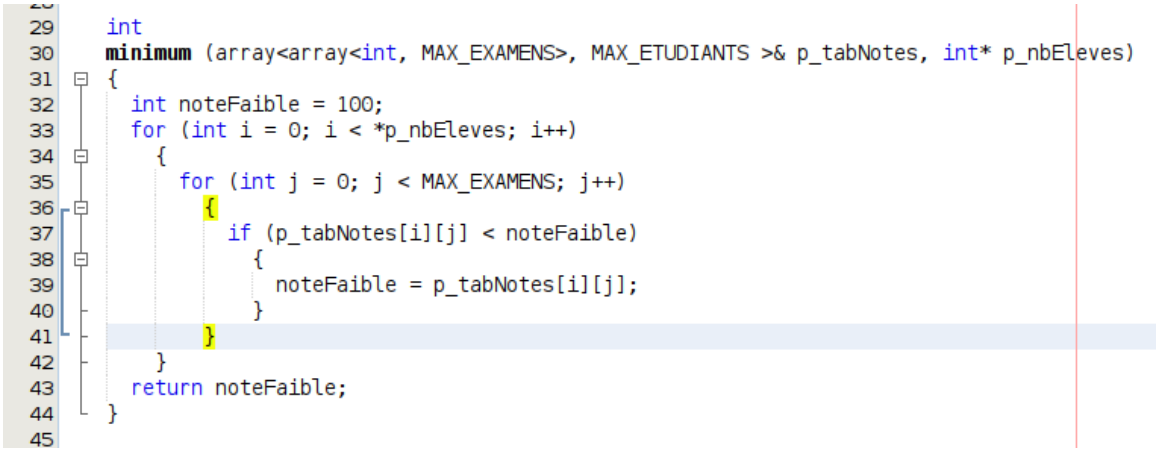
Justification :

La boucle initialisée va tourner autour de l'adresse i.

Explication :

La fonction minimum va devoir nous retourner la valeur minimale d'un examen d'un élève subséquent, et ainsi de suite pour tous les autres élèves. C'est pourquoi on va devoir, tourner la boucle autour de l'adresse j.

Solution :



```
29  int
30  minimum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
31  {
32    int noteFaible = 100;
33    for (int i = 0; i < *p_nbEleves; i++)
34      {
35        for (int j = 0; j < MAX_EXAMENS; j++)
36          {
37            if (p_tabNotes[i][j] < noteFaible)
38              {
39                noteFaible = p_tabNotes[i][j];
40              }
41          }
42      }
43    return noteFaible;
44  }
45
```

Erreur 6 : fichier fonctionsUtilitaires.h ligne 16

Localisation :

```
#ifndef FONCTIONSUTILITAIRES_H_
#define FONCTIONSUTILITAIRES_H_
#include <array>

const int MAX_EXAMENS = 2;
const int MAX_ETUDIANTS = 4;

void saisieNotes (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS > p_tabNotes, int*);
int minimum (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int*);
int maximum (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int*);
double ecartType (std::array<int, MAX_EXAMENS>& p_tabNotesEtudiant);
void afficherTableau (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves);


#endif /* FONCTIONSUTILITAIRES_H_ */

```

L'erreur ici est une erreur de syntaxe, on a oublié de préciser l'adresse vers laquelle le tableau va pointer pour inscrire la note des étudiants ici en l'occurrence p_tabNotes.

Justification :

Ici, on fait seulement un appel à la variable déclarée, on ne connait ni sa valeur ni son adresse. Cette erreur empêche la bonne compilation du code.

Explication :

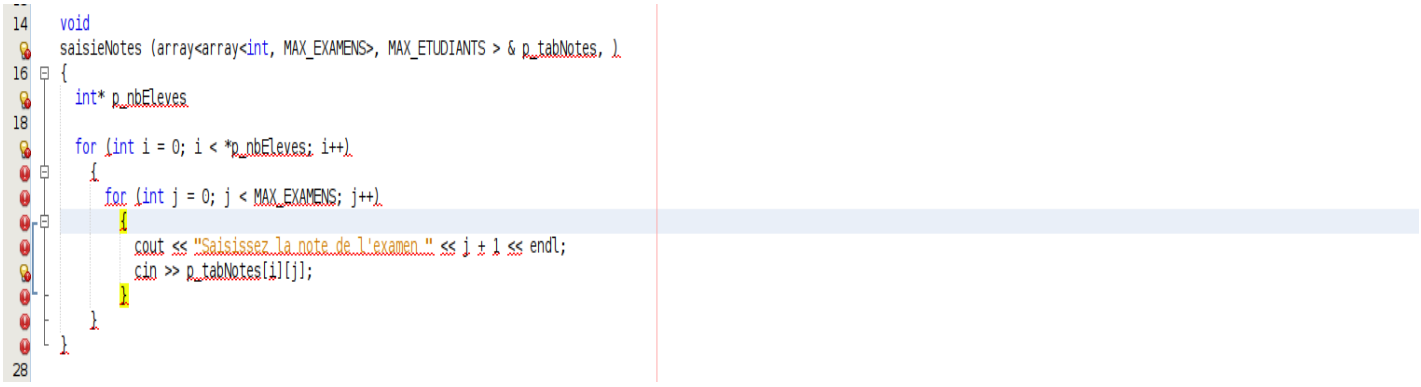
On va devoir donc faire appel aux pointeurs qui sont symbolisés par le symbole & qui réfère à leurs adresses. Afin qu'ici la note saisie pourra rejoindre les autres paramètres sur le tableau, on va devoir donc remplacer p_tabNotes par &p_tabNotes.

Solution :

```

#ifndef FONCTIONSUTILITAIRES_H_
#define FONCTIONSUTILITAIRES_H_
#include <array>

const int MAX_EXAMENS = 2;
const int MAX_ETUDIANTS = 4;

void saisieNotes (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS > &p_tabNotes, int*);
int minimum (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int*);
int maximum (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int*);
double ecartType (std::array<int, MAX_EXAMENS>& p_tabNotesEtudiant);
void afficherTableau (std::array<std::array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves);


#endif /* FONCTIONSUTILITAIRES_H_ */

```

Erreur 7 : fichier fonctionsUtilitaires.cpp ligne 16

Localisation :



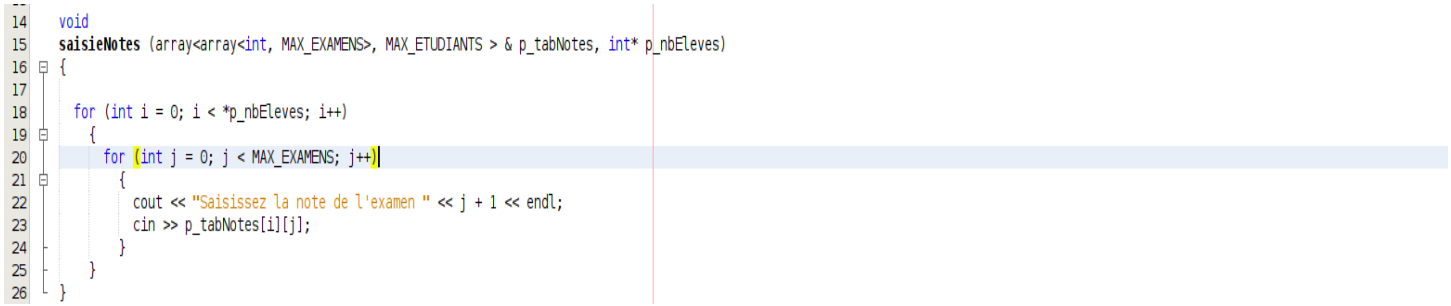
Justification :

C’est une erreur d’édition de lien. Le nombre d’élèves est censé être un argument de la fonction saisieNotes vu que ce dernier constitue une grande importance dans le traçage du tableau. Cette erreur est irrévocablement très pertinente à la compilation du code.

Explication :

De ce fait, on devra donc placer cette variable dans la fonction en tant qu’argument en n’oubliant pas de préciser le type de sa valeur ici en l’occurrence un nombre entier soit « int ».

Solution :



On remarque qu’après l’erreur n’apparait plus sur le terminal de la console.

Erreur 8 : fichier fonctionsUtilitaires.cpp ligne 35

Localisation :

```
29  int
30  minimum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
31  {
32    int noteFaible = 100;
33    for (int i = 0; i < MAX_ETUDIANTS; i++)
34      {
35        for (int j = 0; j < MAX_EXAMENS; j++)
36          {
37            if (p_tabNotes[i][j] < noteFaible)
38              {
39                noteFaible = p_tabNotes[i][j];
40              }
41          }
42      }
43    return noteFaible;
44  }
```

Justification :

MAX_ETUDIANTS est déjà définie et affectée à la valeur de 4, cependant la boucle va être infinie. Vu qu'on a fixé MAX_ETUDIANTS en tant que paramètre dans le tableau des notes, on sera omis à venir modifier cette erreur. Cette erreur constitue une erreur de logique. Donc le compilateur n'aurait jamais pu trouver cette erreur.

Explication :

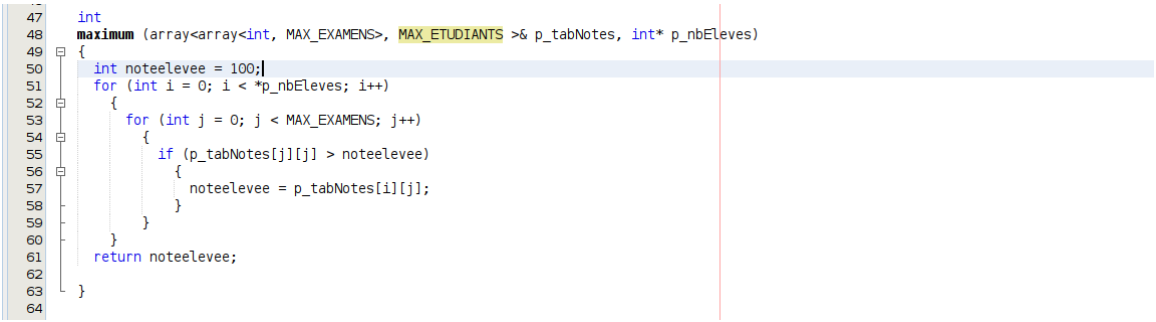
On va devoir remplacer la variable MAX_ETUDIANTS par la variable p_nbEleves qui constitue le nombre des étudiants.

Solution :

```
29  int
30  minimum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
31  {
32    int noteFaible = 100;
33    for (int i = 0; i < *p_nbEleves; i++)
34      {
35        for (int j = 0; j < MAX_EXAMENS; j++)
36          {
37            if (p_tabNotes[i][j] < noteFaible)
38              {
39                noteFaible = p_tabNotes[i][j];
40              }
41          }
42      }
43    return noteFaible;
44  }
```

Erreur 9 : fichier fonctionsUtilitaires.cpp ligne 50

Localisation :



```
47  int
48  maximum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
49  {
50    int noteelevee = 100;
51    for (int i = 0; i < *p_nbEleves; i++)
52      {
53        for (int j = 0; j < MAX_EXAMENS; j++)
54          {
55            if (p_tabNotes[j][j] > noteelevee)
56              {
57                noteelevee = p_tabNotes[i][j];
58              }
59          }
60      }
61    return noteelevee;
62
63  }
64
```

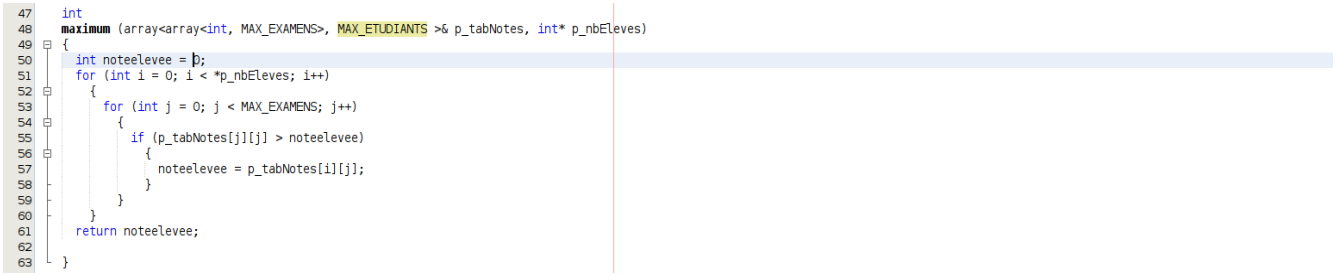
Justification :

C'est une erreur de logique, la note maximale qu'on doit trouver à partir de la fonction « maximum » ne devrait pas être égale à 100. Ce qui va nous retourner à chaque fois, sur le terminal la valeur de 100 comme note maximale.

Explication :

On devrait faire en sorte que la note maximale qui a été déclarée, en haut du bloc de l'énoncé égale à 0 afin que chaque note qui soit le plus grande et qui se trouve à l'adresse correspondante qu'on va retourner va être bel et bien retourner. Par conséquent, après l'exécution de la boucle, la valeur maximale attribuée sera celle qui se trouve à l'emplacement correspondant tel que la boucle le montre d'où le « int notelevee = 0 ».

Solution :



```
47  int
48  maximum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
49  {
50    int noteelevee = 0;
51    for (int i = 0; i < *p_nbEleves; i++)
52      {
53        for (int j = 0; j < MAX_EXAMENS; j++)
54          {
55            if (p_tabNotes[j][j] > noteelevee)
56              {
57                noteelevee = p_tabNotes[i][j];
58              }
59          }
60      }
61    return noteelevee;
62
63  }
```

Erreur 10 : fichier fonctionsUtilitaires.cpp ligne 55

Localisation :

```
45
46
47    int
48    maximum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
49    {
50      int noteelevee = 0;
51      for (int i = 0; i < *p_nbEleves; i++)
52        {
53          for (int j = 0; j < MAX_EXAMENS; j++)
54            {
55              if (p_tabNotes[j][j] > noteelevee)
56                {
57                  noteelevee = p_tabNotes[i][j];
58                }
59            }
60        }
61      return noteelevee;
62
63    }
64
```

Justification :

C’est une erreur de logique. On n’aura pas toujours la bonne valeur maximale retournée.

Explication :

L’erreur commise ici, est une erreur d’indexation plus précisément, on a une erreur ou l’indice est censé indiquer le bon emplacement de la valeur de son contenu soit p_tabNotes[j][j]. C’est pourquoi on va devoir le remplacer par p_tabNotes[j][i].

Solution :

```
47    int
48    maximum (array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS >& p_tabNotes, int* p_nbEleves)
49    {
50      int noteelevee = 0;
51      for (int i = 0; i < *p_nbEleves; i++)
52        {
53          for (int j = 0; j < MAX_EXAMENS; j++)
54            {
55              if (p_tabNotes[j][i] > noteelevee)
56                {
57                  noteelevee = p_tabNotes[i][j];
58                }
59            }
60        }
61      return noteelevee;
62
63    }
64
```

Erreur 11 : fichier fonctionsUtilitaires.cpp ligne 76

Localisation :

```
66   double
67   ecartType (array<int, MAX_EXAMENS>& p_tabNotesEtudiant)
68   {
69     int i = 0;
70     float somme = 0;
71     float sommeCarree = 0;
72
73     while (i < MAX_EXAMENS)
74       {
75         somme += p_tabNotesEtudiant[i];
76         sommeCarree =+ p_tabNotesEtudiant[i] * p_tabNotesEtudiant[i];
77         i++;
78       }
79     float moyenne;
80     moyenne = somme / MAX_EXAMENS;
81     double variance;
82     variance = sommeCarree / MAX_EXAMENS - moyenne * moyenne;
83     double ecartType;
84     ecartType = sqrt (variance);
85     return ecartType;
86   }
87
88
```

Justification :

C'est une erreur de syntaxe. La valeur qu'on attribue à sommeCarree dans ce cas là demeure inconnue, mais cependant sa valeur est attribuée à l'expression qui est à droite ce qui rend l'expression du programme totalement illogique. C'est un type d'erreur difficile à trouver avec le compilateur, d'où l'utilité du débogueur qui nous permet de mettre un breakpoint là ou se situe l'erreur.

Explication :

On va devoir remplacer le « =+ », par un « += »; ce qui sera plus crédible car on incrémentera à la valeur de sommeCarree la valeur de l'expression de droite.

Solution :

```
64
65
66   double
67   ecartType (array<int, MAX_EXAMENS>& p_tabNotesEtudiant)
68   {
69     int i = 0;
70     float somme = 0;
71     float sommeCarree = 0;
72
73     while (i < MAX_EXAMENS)
74       {
75         somme += p_tabNotesEtudiant[i];
76         sommeCarree += p_tabNotesEtudiant[i] * p_tabNotesEtudiant[i];
77         i++;
78       }
79     float moyenne;
80     moyenne = somme / MAX_EXAMENS;
81     double variance;
82     variance = sommeCarree / MAX_EXAMENS - moyenne * moyenne;
83     double ecartType;
84     ecartType = sqrt (variance);
85     return ecartType;
86   }
87
88
```

Erreur 12 : fichier programmePrincipal.cpp ligne 17

Localisation :

```cpp
/**
 * \file programmePrincipal.cpp
 * \brief
 */

#include <iostream>
#include <array>
#include "fonctionsUtilitaires.h"


using namespace std;


int
main ()
{
  int num;
  int nbEleves;
  array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS> tabNotes;

  bool valeurCorrecte = false;
  while (!valeurCorrecte)
    {
      cout << "Saisissez le nombre d'élèves" << endl;
      cin >> nbEleves;
      if (nbEleves > 0 && nbEleves <= MAX_ETUDIANTS)
        {
```

Justification :

On remarque que tout au long du code du fichier programmePrincipal.cpp; on n'a pas utilisé la variable num qui a été déclarée tout en haut. C'est une mauvaise pratique.

Explication :

Vu qu'on n'utilise pas cette variable, mieux vaut l'enlever. La supprimer ne fera aucun mal au code.

Solution :

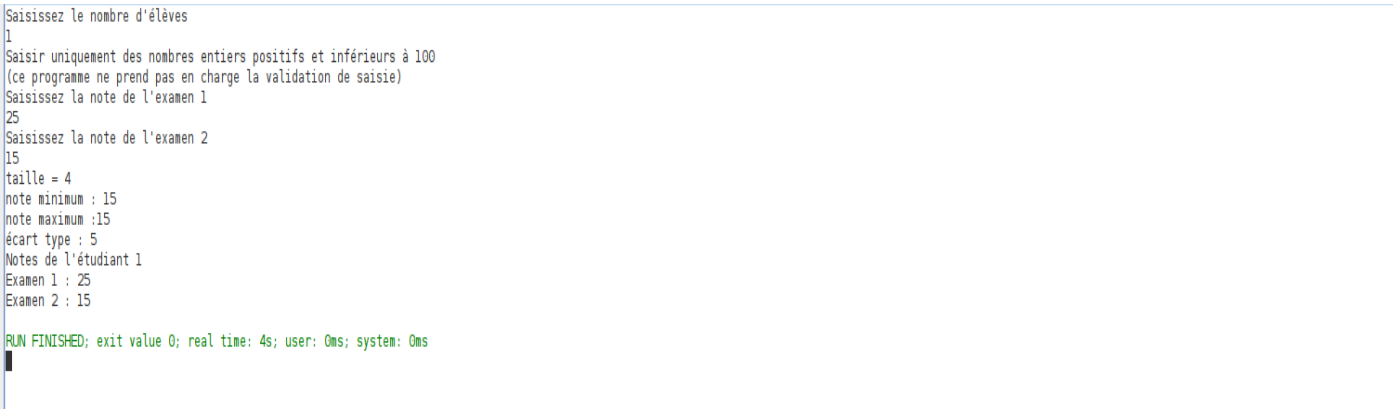
```cpp
int
main ()
{

  int nbEleves;
  array<array<int, MAX_EXAMENS>, MAX_ETUDIANTS> tabNotes;

  bool valeurCorrecte = false;
  while (!valeurCorrecte)
    {
      cout << "Saisissez le nombre d'élèves" << endl;
      cin >> nbEleves;
      if (nbEleves > 0 && nbEleves <= MAX_ETUDIANTS)
        {
          valeurCorrecte = true;
```

Allure du code retourné :

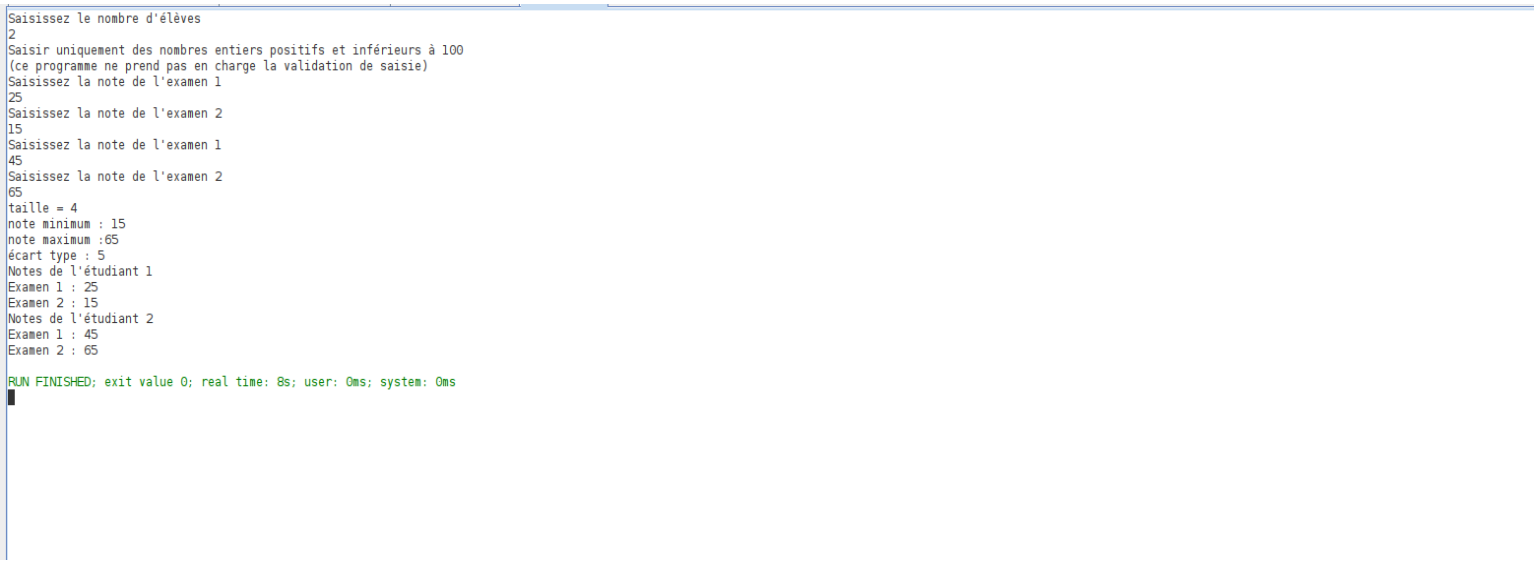
Après avoir corriger toutes les erreurs, on a exécuté le code et tenté plusieurs scénarios pour vérifier que le code fonctionne.

Et voilà ce que nous affiche la console :

1er Scénario :



2ème Scénario :



Effectivement, le code se compile parfaitement comme convenu, et nous retourne les valeurs exactes dans leurs cases correspondantes.