**TP3 Gestion de Projets informatiques**

Exo1:

public static void tri\_insertion\_elementsPositifs(ref int[] tab, int taille)

{

int[] newTab = new int[taille];

int i;

for (i = 0; i < taille; i++) newTab[i] = -1;

for (i = 1; i < taille; i++)

if (tab[i] >= 0)

newTab[i] = tab[i];

else newTab[i] = 0;

tab = newTab;

int j;

for (i = 1; i < taille; i++)

{

int elem = tab[i];

for (j = i; j > 1 && tab[j - 1] > elem; j--)

tab[j] = tab[j - 1];

tab[j] = elem;

}

}

**2.1.1-------------------------------------------------------------------------------------------------------------------------**

|  |
| --- |
| **CC = E – N +2**  **CC = 19 – 15 + 2**  **CC = 6** |

* Le nœud 5’ concerne la ligne le code de la boucle for qui est sur la même ligne, et 7’ est nœud fictif qui concerne soit la sortie de « if » ou « else ».

**2.1.2-------------------------------------------------------------------------------------------------------------------------**

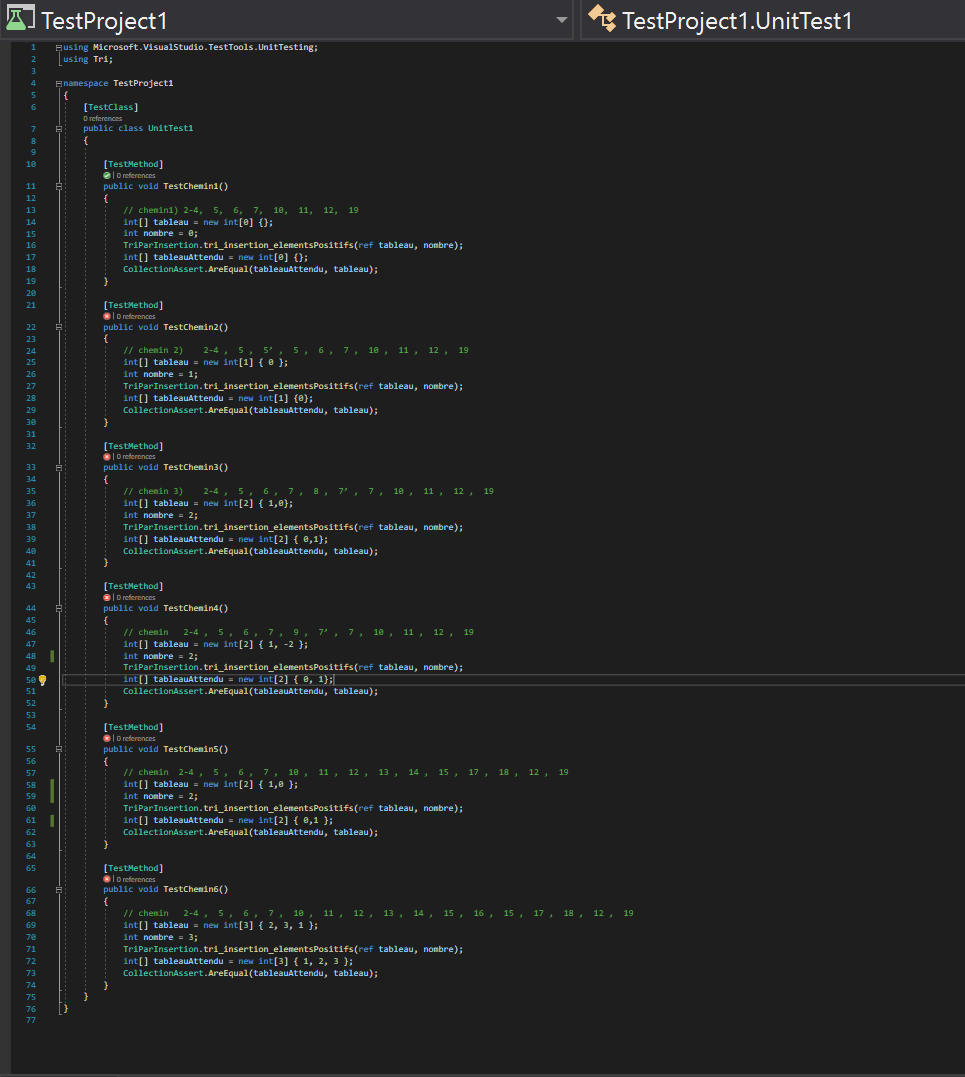
Les chemins linéairement indépendants couvrants toutes les instructions sont :

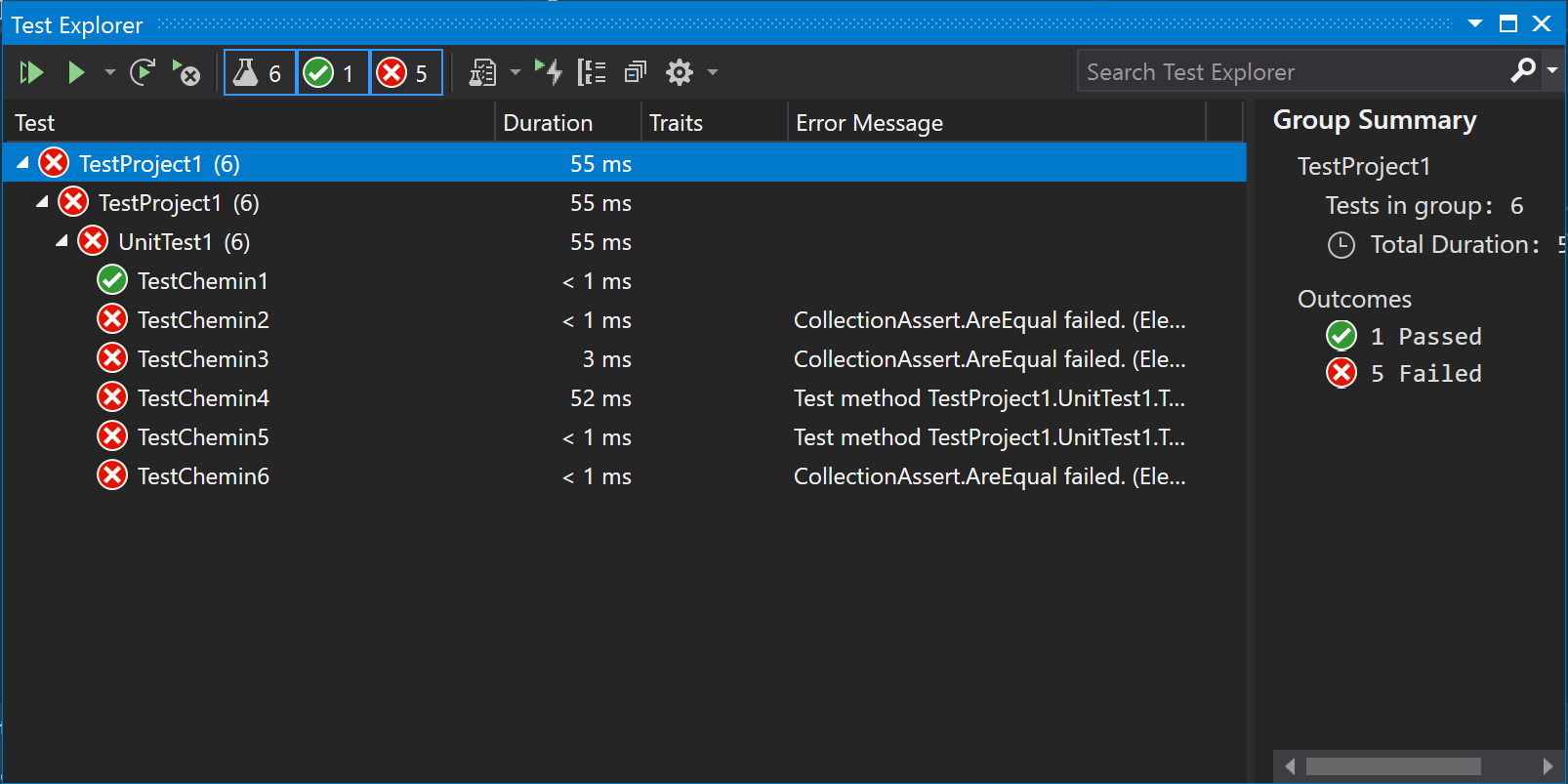
1. 2-4 🡪5🡪6 🡪7 🡪10 🡪11 🡪12 🡪19
2. 2-4 🡪5 🡪5’ 🡪5 🡪6 🡪7 🡪10 🡪11 🡪12 🡪19
3. 2-4 🡪5 🡪6 🡪7 🡪8 🡪7’ 🡪7 🡪10 🡪11 🡪12 🡪19
4. 2-4 🡪5 🡪6 🡪7 🡪9 🡪7’ 🡪7 🡪10 🡪11 🡪12 🡪19
5. 2-4 🡪5 🡪6 🡪7 🡪10 🡪11 🡪12 🡪13 🡪14 🡪15 🡪17 🡪18 🡪12 🡪19
6. 2-4 🡪5 🡪6 🡪7 🡪10 🡪11 🡪12 🡪13 🡪14 🡪15 🡪16 🡪15 🡪17 🡪18 🡪12 🡪19

**2.1.3-------------------------------------------------------------------------------------------------------------------------**





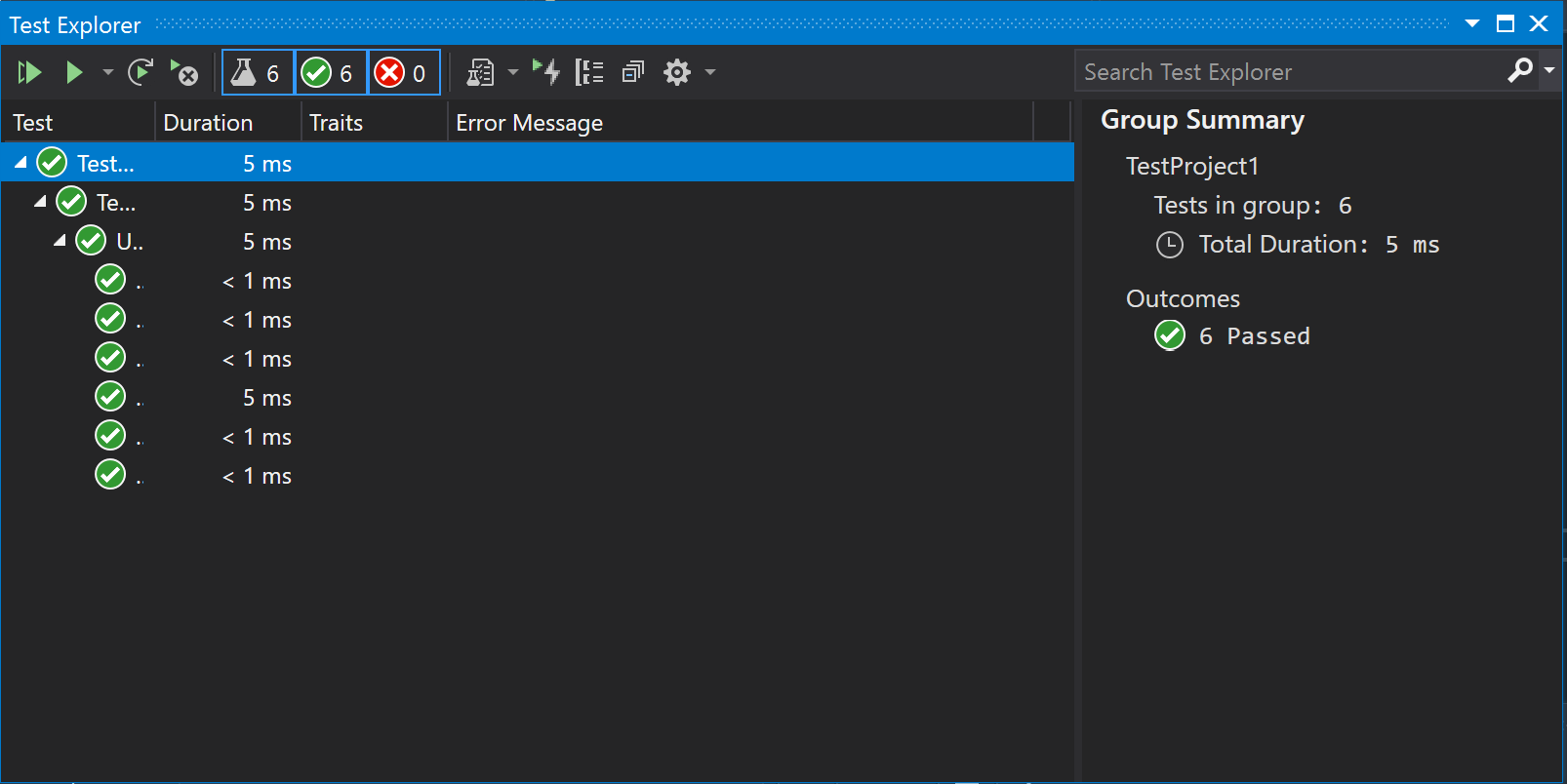


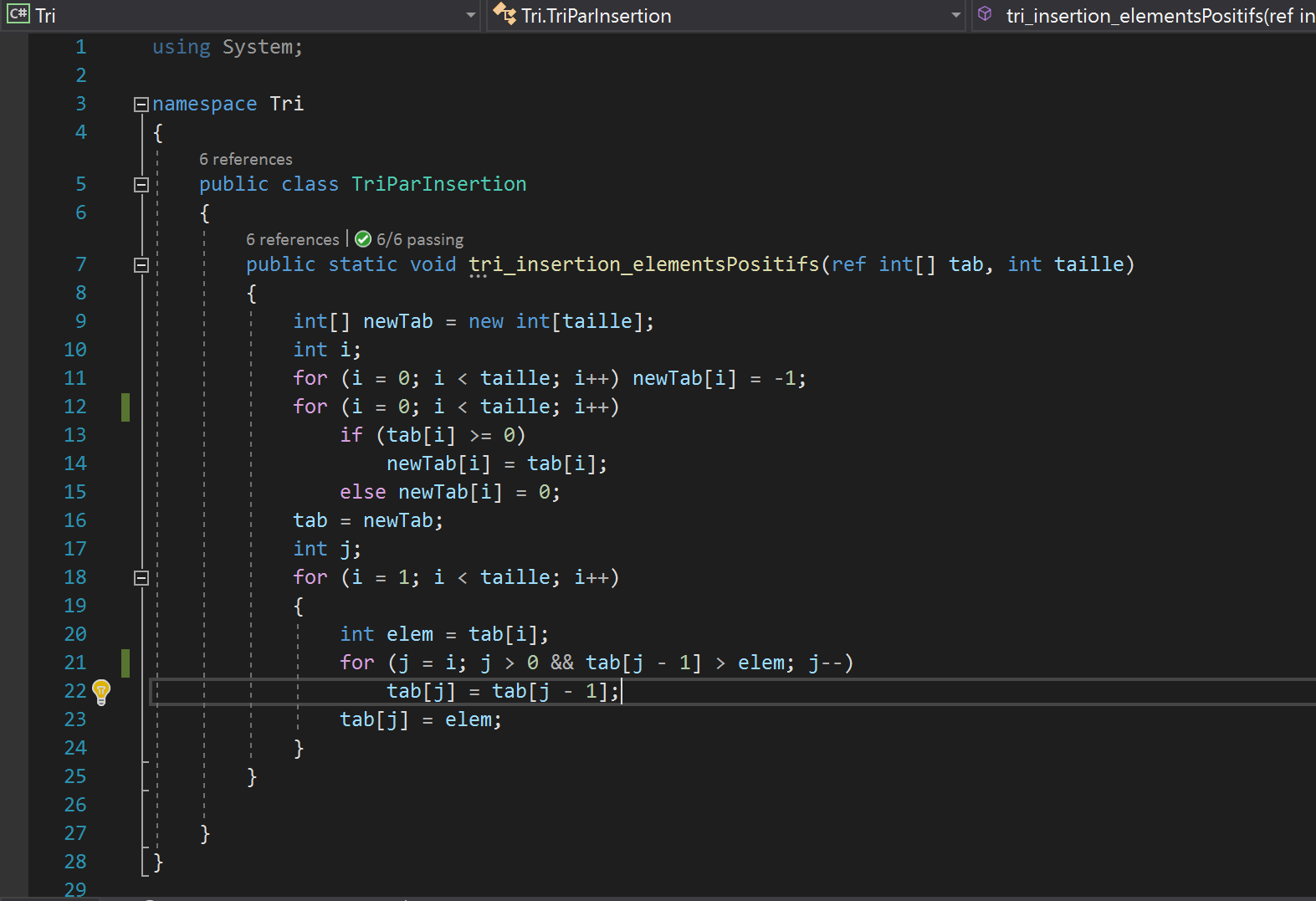


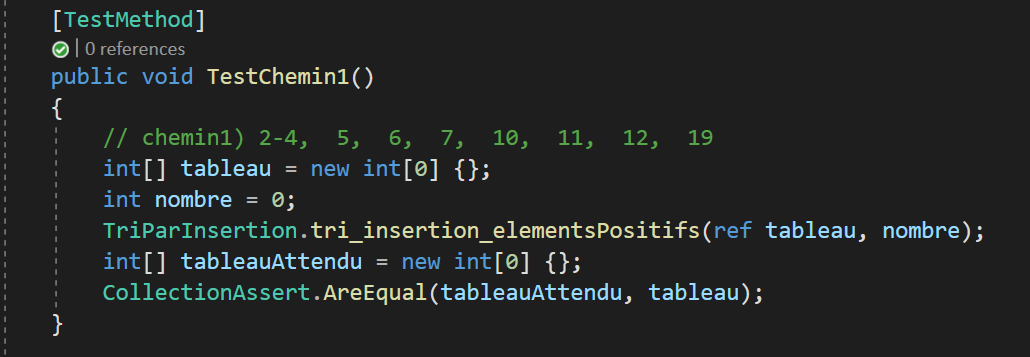
**2.1.4-------------------------------------------------------------------------------------------------------------------------**

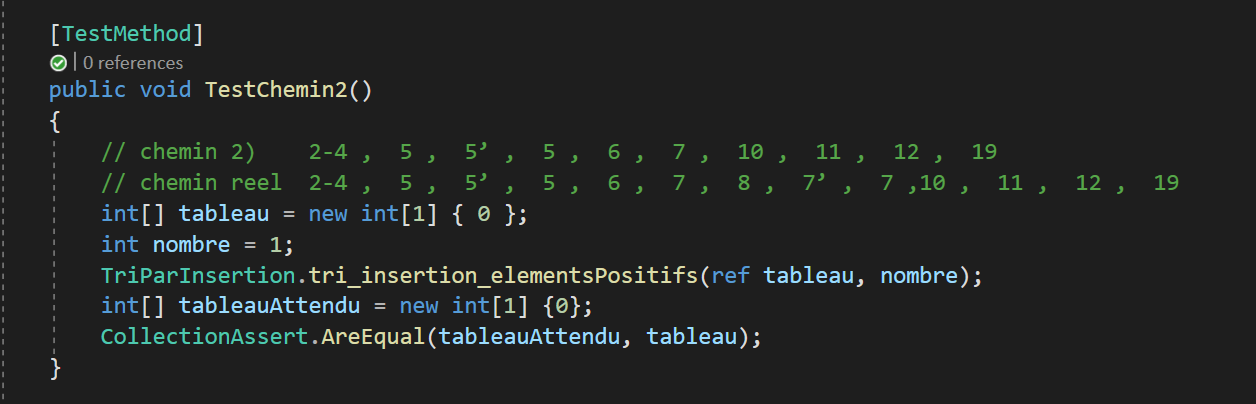
L’origine de l’erreur comme le montre la première capture est le point de départ de la deuxième boucle qui doit être 0 au lieu de 1, ainsi que la condition d’entrée de la dernière boucle qui doit être j>0 au lieu de j>1 pour ne pas exclure le tri pour les deux premiers éléments de la liste.

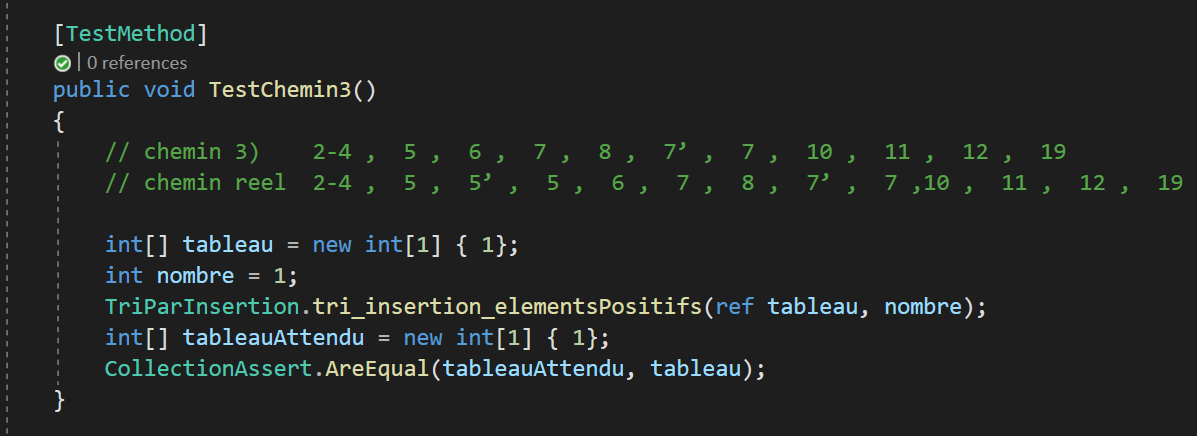
Après avoir effectué les modifications nécessaires, on fait alors un test de regression, on obtient par la suite un succès pour tous les tests.

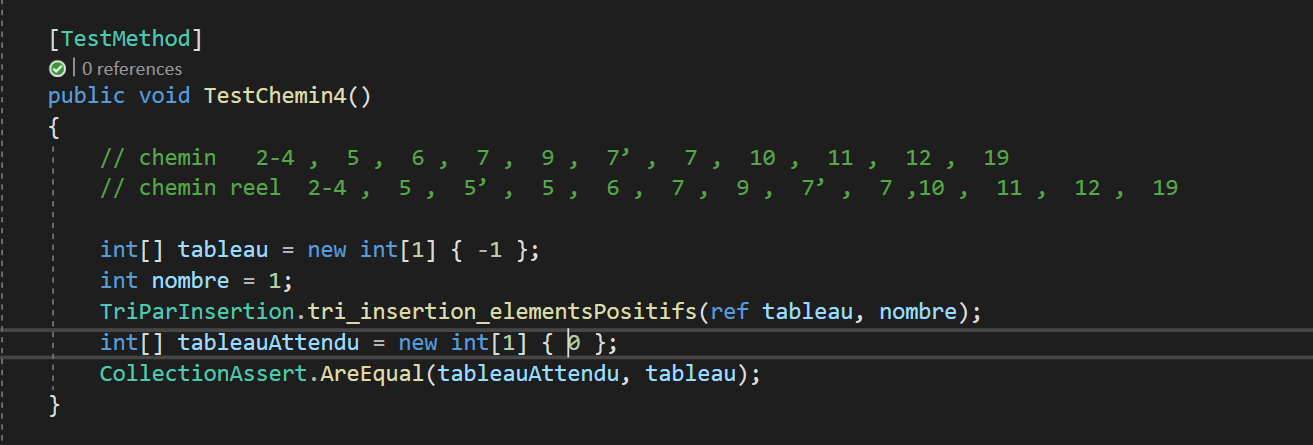


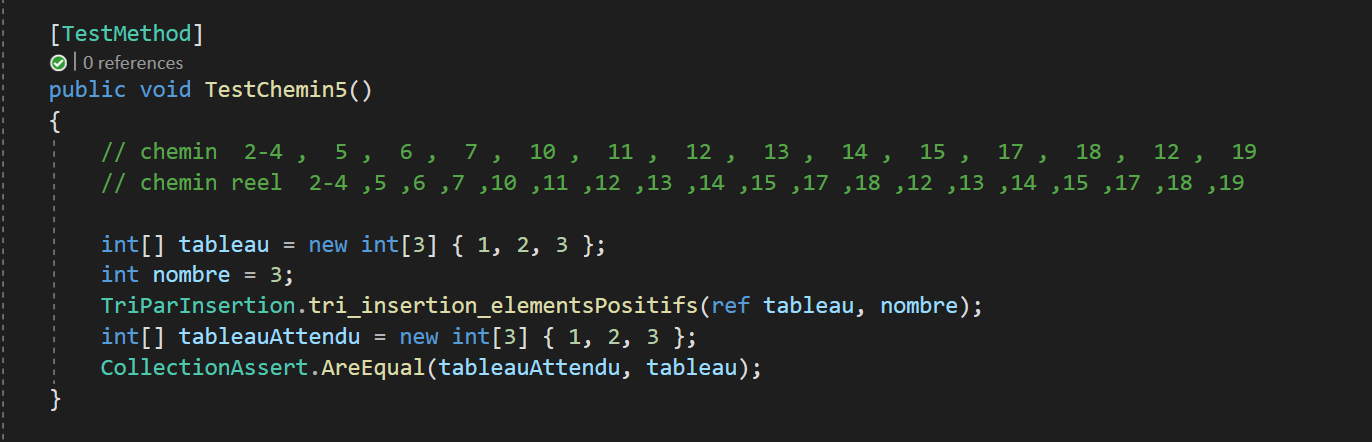


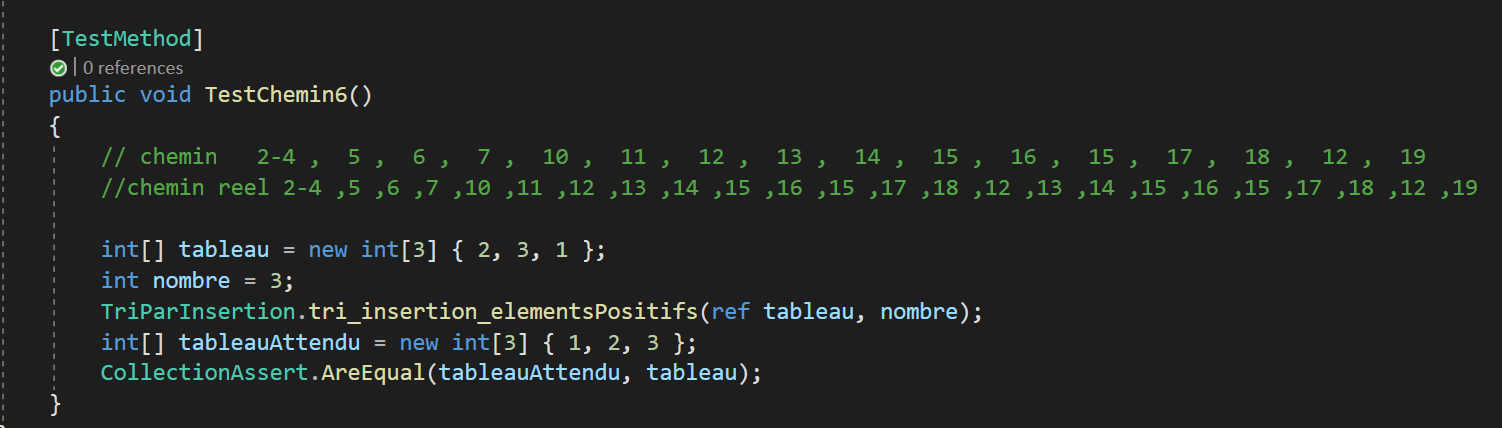












**2.2 – Test fonctionnel :**

**2.2.1-------------------------------------------------------------------------------------------------------------------------**

**Selenium Webdriver**

**Définition :** Selenium WebDriver est une collection d'API open source qui sont utilisées pour automatiser le test d'une application Web.

L'outil Selenium WebDriver est utilisé pour automatiser les tests d'applications Web afin de vérifier qu'il fonctionne comme prévu. Il prend en charge de nombreux navigateurs tels que Firefox, Chrome, IE et Safari. Cependant, en utilisant Selenium WebDriver, nous pouvons automatiser les tests pour les applications Web uniquement. Il n'est pas éligible pour les applications basées sur les fenêtres. Il prend également en charge différents langages de programmation tels que C #, Java, Perl, PHP et Ruby pour l'écriture de scripts de test. Selenium Webdriver est indépendant de la plate-forme car le même code peut être utilisé sur différents systèmes d'exploitation tels que Microsoft Windows, Apple OS et Linux. C'est l'un des composants de la famille du sélénium, qui comprend également Selenium IDE, Selenium Client API, Selenium Remote Control et Selenium Grid.

Selenium WebDriver ne gère pas le composant de fenêtre, mais cette limitation peut être surmontée en utilisant des outils externes tels que l'outil AUTO IT, Sikuli, etc. Sélecteur CSS et Xpath. Il prend également en charge les pages Web dynamiques comme Ajax, où les éléments de la page Web peuvent changer sans que la page elle-même ne soit rechargée. En utilisant différents fichiers jar, nous pouvons également tester l'API, le test de base de données, etc. à l'aide de Selenium WebDriver.

**Différence entre Selenium RC et Webdriver :**

Avant l'avènement de WebDriver en 2006, il existait un autre outil d'automatisation appelé Selenium Remote Control. WebDriver et Selenium RC présentent les fonctionnalités suivantes :

* Ils permettent tous deux d'utiliser un langage de programmation dans la conception de scripts de test.
* Ils permettent tous deux d'exécuter vos tests sur différents navigateurs.
* **L'architecture de WebDriver est plus simple que celle de Selenium RC**.
* Il contrôle le navigateur à partir du niveau du système d'exploitation
* Il est besoin juste de l'IDE du langage de programmation (qui contient les commandes Selenium) et un navigateur.
* **WebDriver est plus rapide que Selenium RC car il** parle directement au navigateur utilise le propre moteur du navigateur pour le contrôler.
* **WebDriver interagit avec les éléments de la page de manière plus réaliste.** Par exemple, si on a une zone de texte désactivée sur une page qu’on teste, WebDriver ne peut vraiment pas y entrer de valeur, tout comme une personne réelle ne le peut pas.
* **WebDriver peut prendre en charge le navigateur HtmlUnit** qui est qualifié de "headless" car il s'agit d'un navigateur invisible - il est sans interface graphique.

C'est un navigateur très rapide car aucun temps n'est passé à attendre le chargement des éléments de la page. Cela accélère les cycles d'exécution de test.

Puisqu'il est invisible pour l'utilisateur, il ne peut être contrôlé que par des moyens automatisés.

**Browser stack (Selinum)**

BrowserStack est une plate-forme de test Web et mobile dans le cloud qui permet de tester les sites Web et les applications mobiles sur des navigateurs à la demande, des systèmes d'exploitation et de vrais appareils mobiles, sans exiger que les utilisateurs installent ou maintiennent un laboratoire interne de machines virtuelles, d'appareils ou d'émulateurs.

Les utilisateurs peuvent choisir parmi plus de 1 200 appareils mobiles, navigateurs et systèmes d'exploitation réels à la demande et s'appuyer sur une infrastructure sécurisée, stable et évolutive pour prendre en charge des milliers de tests manuels et automatisés simultanés. En réduisant le temps, les coûts et les frais de maintenance associés aux tests, BrowserStack aide les équipes à rester concentrées sur ce qui compte le plus : créer de meilleurs produits et services.

**Enregistrement Video :**

Lien Google Drive:

<https://drive.google.com/file/d/16Lf10NKXh-sqYWWzk7ow4KkYAiV5OrX3/view?usp=sharing>

**Code de l’application :**

Lien Google Drive :

<https://drive.google.com/file/d/1Aiecyie_5i6XrK_n18W4_9aGg1iDtWc5/view?usp=sharing>

GitHub :

<https://github.com/AYCH17/simple-express-angular-app>

**Références :**

<https://www.selenium.dev/documentation/en/webdriver/>

<https://www.browserstack.com/guide/selenium-webdriver-tutorial>

<https://www.browserstack.com/selenium>

<https://fr.acervolima.com/applications-et-utilisations-de-selenium-webdriver/>