**Software Engineering (10 points)**

Pour les questions 1 et 2, veuillez répondre directement dans le fichier texte. Pour les questions 3 et 4, des fichiers vous sont fournis, veuillez remettre vos fichiers réponse dans une archive zip.

**Question 1 (0.5 point)**

Laquelle des options suivantes décrit le mieux les règles du TDD ?

Option a)

1. Écrire du code exempt de mauvaises odeurs
2. Écrire un test unitaire
3. Vérifier que le test passe, sinon modifier le code

Option b)

1. Écrire un test unitaire
2. Vérifier que le test échoue
3. Écrire le code minimal pour faire passer le test
4. Refactor le code tout en s’assurant que le test passe toujours

Option c)

1. Écrire les tests unitaires
2. S’assurer que les tests passent
3. Écrire le code le plus structuré possible
4. S’assurer que les tests passent toujours

Option d)

1. Écrire du code exempt de mauvaise odeur
2. Écrire un test unitaire
3. Vérifier que le test passe, sinon modifier le test

Réponse : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 2 (2.5 points)**

1. Qu’est-ce qu’une branche dans Git? Quels sont les principaux avantages des branches ?

1. Est-ce possible d’annuler un commit ayant déjà été pushé? Si oui comment ? sinon pourquoi ?

1. Comment peut-on voir les modifications apportées dans un commit en particulier ?

1. Qu’est-ce qu’un conflit dans Git? Quelle est la procédure pour résoudre les conflits?

1. Écrivez les commandes git pour les fonctionnalités suivantes :
   1. Créer une copie locale du répertoire distant <https://cs.git.com/repository>.

* 1. Créer une nouvelle branche feature/action et aller sur cette branche.

* 1. Après avoir modifier le fichier main.js, envoyer ces modifications sur le repo distant.

* 1. Vous vous trouvez toujours sur la branche feature/action, vous voulez fusionner cette branche à la branche develop.

**Question 3 (3 points)**

Le code du fichier question3.ts présente des problèmes de qualité et a besoin de refactoring. Indiquez les lignes où il y a des mauvaises odeurs à l’aide d’un commentaire à la fin de la ligne. Dans votre commentaire, indiquez quel est le problème et expliquez très brièvement comment ce problème pourrait être réglé. Si des problèmes sont présent dans le code en général ou sur plusieurs lignes, veuillez le noter dans un commentaire au bas du fichier)

**Question 4 (4 points)**

On vous demande de développer la fonctionnalité utilisée pour inscrire les participants à la compétition de jeux vidéo aux CS Games 2019. Les exigences sont les suivantes :

- Le nom d’utilisateur du participant sera formé de son prénom, son nom ainsi que de son année de naissance concaténés l’un à la suite de l’autre. Ex : AvrilLavigne1984. Les caractères spéciaux ne sont pas permis.

- Une fois le nom d’utilisateur créé, le joueur est ajouté à la liste de joueur déjà inscrit.

- Le nombre maximal de joueurs inscrits est de 100/x arrondi vers le haut et où x est le nombre d’universités anglophones participant aux CS Games. Pour l’instant x demeure inconnu, car les inscriptions n’ont pas encore été faites.

Comme tout bon développeur respectant les conventions, vous écrivez vos tests avant d’écrire vos fonctions.

1. Fournissez le code des tests unitaires qui vous permettront de tester vos fonctions. (\*) (2.5 points)
2. Écrivez ensuite vos fonctions de manière à ce que vos tests passent. (\*)   
   (1.5 point)

(\*) Notez bien qu’étant donné le court temps qui vous est accordé pour l’examen, nous ne vous demandons pas de configurer un environnement de tests ni d’utiliser un framework de tests unitaires. Il s’agit plutôt ici de vérifier que vous avez une bonne structure générale de tests qui concorde bien avec le code que vous aurez écrit.