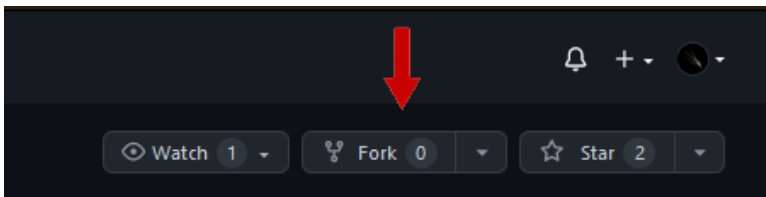


Application Pull Request - Etudiant C

Suivez cette partie avec les autres étudiants:

1. Si ce n'est pas déjà fait, créez un compte GitHub (<https://github.com/signup>)
2. Attendez que l'Etudiant A vous fournisse son nom d'utilisateur GitHub ou l'url du projet qu'il vient de créer.
3. Trouvez le projet "syracuse_project" récemment publié par votre collègue sur son profil
4. Faites un fork du dépôt vers votre profil:

En haut à droite de la page du projet de l'Etudiant A, cliquez sur le bouton "Fork"



(Pour la suite, ne montrez pas vos consignes aux autres étudiants)

A faire de votre côté exclusivement:

5. Maintenant que vous avez créé un fork du projet de l'étudiant A, vous pouvez cloner votre Fork en local sur votre poste (l'url qui contient votre nom d'utilisateur)

```
1 git clone [url de votre fork]
```

Inspectez le code une fois que vous aurez cloné le projet.

6. Vous allez aussi pouvoir intégrer un lien vers le dépôt originel:

```
1 git remote add upstream [url repo etudiant A]
```

`upstream` est le nom de remote communément utilisé pour désigner la remote source d'un projet duquel vous dépendez. Ce lien vous permet de suivre les modifications qui pourraient être intégrées sur la branche principale du projet.

En utilisant la commande suivante, vous pouvez voir les remote configurées sur votre projet:

```
1 git remote -v
```

Par défaut, un projet cloné depuis GitHub possède la remote origin qui désigne la source initiale de votre travail.

7. L'algorithme de Syracuse que l'étudiant A vient de publier présente une multitude de trous à compléter:

Vous allez compléter **uniquement le cas PAIR**.

(Créez une nouvelle branche "add_even" pour ajouter cette fonctionnalité)

MAIS: Intégrez une erreur grossière lors de votre ajout :) ou alors, ajoutez un fichier aléatoire dans l'arborescence du projet.

8. Publiez votre nouvelle branche modifiée vers votre propre version sur GitHub

```
1 git push origin add_even
```

Puisque votre branche n'existe pas encore sur le serveur cible, git devrait se plaindre et vous fournir une commande adaptée afin de créer la nouvelle branche cible. La nouvelle commande devrait ressembler à:

```
1 git push --set-upstream origin add_even
```

9. Lorsque vous revenez sur votre dépôt personnel, un bandeau doit apparaître, mentionnant qu'une nouvelle branche possède des modifications récentes.

Réalisez une Pull Request vers le dépôt de l'Etudiant A à partir de cette nouvelle branche

-
10. Attendez que l'Etudiant A, intervienne sur votre Pull Request.

Espérons qu'il y voit l'erreur que vous y avez introduit... Si c'est le cas, il vous l'a indiqué sur la page de votre Pull Request. Sinon, je vous laisse le soin de lui indiquer (poliment) que vous avez introduit une erreur volontairement et qu'il aurait dû y prêter attention.

11. Corrigez l'erreur depuis la même branche que vous avez utilisé jusqu'à maintenant. Et publiez cette correction sur votre dépôt de la même manière que précédemment. A vous de voir, si vous souhaitez faire un nouveau commit, ou un amend.

(L'utilisation d'amend n'est pas recommandé puisque vous allez réécrire l'histoire. Dans ce cas, le serveur Github va vous interdire l'écriture, même sur votre propre dépôt. Pour forcer l'écriture, utilisez l'argument `-f` soit: `git push -f`)

Note: Puisque votre branche `add_even` fait actuellement l'objet d'une Pull Request, tous les commits que vous publierez sur cette branche seront automatiquement ajoutés à la PR.

-
12. Une fois votre Pull Request acceptée et fusionnée, revenez dans votre environnement de travail et tirez les modifications depuis le remote `upstream`:

```
1 git pull upstream main
```

13. Les modifications sur le projet cible ont maintenant été prises en compte.

Inspectez le fichier que vous avez maintenant en local.

Notez que vos modifications n'ont pas été les seules à avoir été ajoutée au projet.

L'étudiant A (et C) ont également pu travailler sur le script. Vous pouvez valider que le script fonctionne correctement depuis le terminal Matlab:

```
1 syracuse(chiffre)
```

14. Vous pouvez maintenant supprimer votre branche local et celle distante:

```
1 git branch -f add_even
2 git push origin --delete add_even
```

15. Essayez maintenant de collaborer simultanément pour rédiger une documentation pour l'algorithme dans le fichier README.md

Vous pouvez vous aider de <https://www.markdownguide.org/basic-syntax/>