

Application Pull Request - Etudiant A

Suivez cette partie avec les autres étudiants:

1. Si ce n'est pas déjà fait, créez un compte GitHub (<https://github.com/signup>)
2. Copiez le dossier "syracuse_project" présent sur le réseau
(Q:\5A\0-Transport Connecté Intelligent (TCI)\Gestion de Versions)
3. Initialisez vous-même le projet sous Git
4. Faites un premier commit du projet dans son état actuel et publiez-le sur GitHub:

Dans un premier temps, créer un nouveau dépôt (publique) sur Github (<https://github.com/new>) qui portera le nom "syracuse_project". Vous pouvez également ajouter une description si vous le souhaitez, mais laissez vides le restant des champs.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner * Repository name *

PYBrlin ▾ / syracuse_project ✓

Great repository names are short and memorable. Need inspiration? How about [refactored-tribble?](#)

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

ⓘ You are creating a public repository in your personal account.

Create repository

5. Donnez votre nom d'utilisateur GitHub ou simplement l'url du dépôt à Etudiant B (et Etudiant C) afin qu'il puisse trouver votre dépôt.

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

A faire de votre côté exclusivement:

6. L'algorithme de Syracuse présent dans le dossier que vous venez de copier présente une multitude de trous à compléter:

Vous allez compléter **uniquement le cas de RECURSIVITE**

Vous pouvez réaliser cet ajout directement sur votre branche "main", puisque vous êtes le mainteneur du projet, vous êtes en droit d'y travailler directement. Faites vos modifications localement, mais attendez avant de pousser vos modifications que les autres étudiants aient cloné votre dépôt sur leurs machines.

7. (Si pas d'Etudiant C) Complétez également **le cas PAIR**

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

1. Publiez votre nouvelle branche modifiée vers votre fork 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
```

origin est le nom de remote par défaut utilisé pour désigner la remote d'origine d'un projet que vous clonez directement.

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

Réalisez une Pull Request depuis votre dépôt, en sélectionnant la branche "main" en tant que destination

Attendez que l'Etudiant B (et l'Etudiant C) revienne(nt) vers vous.

8. Vous venez de recevoir une (ou deux) Pull Requests, inspectez les lignes qui ont été modifiés.

Vous avez des commentaires à faire sur le code ? Ajoutez un commentaire sur la page de la Pull Request en mentionnant l'auteur afin qu'il procède à des modifications.

9. Si les modifications apportées par la Pull Request vous conviennent, cliquez sur "Merge" en bas de la page de la PR.

-
10. Tirez les dernières modifications depuis votre environnement de travail local.

```
1 git pull origin main
```

11. Les modifications en compte. Inspectez le fichier que vous avez maintenant en local.

Notez que toutes les modifications simultanées ont été ajoutée au projet. Vous pouvez valider que le script fonctionne correctement depuis le terminal Matlab:

```
1 syracuse(chiffre)
```

12. (Si pas d'Etudiant C) Vous pouvez maintenant supprimer votre branche "add_even" local et distante (si elle n'a pas déjà été supprimée par l'application de la Pull Request):

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

13. Visualisez l'arborescence des commits du projet en utilisant la commande suivante:

```
1 git log --graph -all [--oneline]
```

14. 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/>