

My projects (/)

(https://signin.intra.42.fr/users/sign_out)



tra.42.fr) (/projects/list)

Your projects

C Exam - Beginner
([projects-exam-beginner](#))



Electronics
(/projects/electronics)
a.42.fr
Libft (/projects/libft)

Rushes (/projects/rushes)



'shop.intra.42.fr)

You should correct 1 student in this team



Git repository

vogsphere@vogsphere.42.fr:intra/2015/activities/lem_in/facorreg

Introduction

Ce barème contient également des zones où VOUS devez décider, selon votre propre expérience, de quelle façon doit être contrôlée une fonctionnalité demandée.

Nous vous demandons, pour le bon déroulement de cette notation :

- de rester courtois, poli, respectueux, constructif, lors de cet échange. Le lien de confiance entre la communauté 42 et vous en dépend.
- de bien mettre en évidence auprès de la personne notée (ou du groupe) les dysfonctionnements éventuels.
- d'accepter qu'il puisse y avoir parfois des différences d'interprétation sur les demandes du sujet ou l'étendue des fonctionnalités. Restez ouvert d'esprit face à la vision de l'autre (a-t-il raison ? tort ?), et notez le plus honnêtement possible.

Bonne soutenance à tous !

Guidelines

RAPPELEZ VOUS QUE VOUS NE DEVEZ CORRIGER QUE CE QUI SE TROUVE SUR LE DÉPÔT DE RENDU DE L'ÉTUDIANT.

Il s'agit de faire un "git clone" du dépôt de rendu, et de corriger ce qui s'y trouve.

Ratings

Define the type of error (if there is an error), which ended the correction.

✓ Ok

Empty work

Incomplete work

No author file

Invalid compilation

Norme

Cheat

Crash

Attachments

Subject (<https://cdn.intra.42.fr/pdf/pdf/185/lem-in.fr.pdf>)

Sections

Préliminaires

Conditions de base

Lecture sur l'entrée standard

Tests d'erreur

Si l'un de ces points est vrai, alors la soutenance est immédiatement terminée.

- Pas de rendu (= rien dans le dépôt git)

- Fichier auteur absent ou invalide

- Faute de norme, la norminette faisant foi

- Cas de triche (l'étudiant utilise une fonction interdite, ou ne sait pas expliquer son code) => cocher Cheat

Yes

No

Lecture sur l'entrée standard

Le programme de l'utilisateur doit lire une fourmière sur l'entrée standard. (Par exemple : ".lem-in < fourmiere.txt")

Si ce n'est pas le cas, le projet est hors-sujet, la correction s'arrête.

Yes

No

Tests d'erreur

Le programme doit gérer les erreurs dans l'entrée de façon appropriée. Une certaine tolérance est de mise quant au choix de traitement fait par l'étudiant, et bien évidemment au contenu de ses messages d'erreurs. Notez l'esprit, pas la petite bête. Comprendre par là que si l'étudiant affiche autre chose que "ERROR", ou s'il choisit par exemple de dire que si la fourmière n'a pas de fourmis, alors ce n'est pas une erreur juste une solution vide ... Eh bien, pourquoi pas. C'est une interprétation valable.

Aucune salle

Gestion du cas d'erreur où il n'y a aucune salle.

Yes

No

Aucune fourmi

Gestion du cas d'erreur où il n'y a aucune fourmi.

Yes

No

Aucun commentaire obligatoire

Gestion du cas d'erreur où il y a absence de ##start / ##end.

Yes

No

https://projects.intra.42.fr/projects/lem_in/evaluations/59/scales/103/scale_teams/232883/edit

2/4

Aucune chemin possible

Gestion du cas d'erreur où il n'y a aucun chemin possible.

☒ Yes☐ No

Gestion des commentaires

Commentaires

Le programme accepte les fourmilières contenant des commentaires.

☒ Yes☐ No

Commandes

Le programme accepte les fourmilières contenant des commandes autre que ##start et ##end .

☒ Yes☐ No

Format de sortie

Le format de sortie doit être respecté.

Composition de la fourmilière

La composition de la fourmilière est répétée sur la sortie.

☒ Yes☐ No

Commandes et commentaires

Les commandes et commentaires sont répétés sur la sortie.

☒ Yes☐ No

Chemin des fourmis

Le chemin est au bon format (1 ligne par tour, N déplacements par tour, un déplacement au format "Lx-y" x étant un numéro de fourmi et y une salle).

☒ Yes☐ No

Justesse de l'algo

Validité du chemin

Le programme doit sortir des chemins valides, mais pas forcément les plus efficaces (pour l'instant, cela vient dans la partie bonus).
À vous de faire un nombre de tests que vous estimez suffisants, avec des entrées valides, et de vérifier à chaque fois que la solution sortie par l'étudiant est valide.
Vérifiez surtout qu'il n'y a pas de fourmis qui se marchent dessus, et que toutes les fourmis sont bien dans la salle ##end à la fin de la solution.

Si aucun de vos tests n'obtient une solution fausse, cochez Oui.

Le programme donne toujours une solution juste ?

☒ Yes☐ No

Bonus

D'autres bonus

Vous pouvez lister ici jusqu'à cinq bonus que vous jugez utiles et bien implémentés.

Exemples de bonus :

- algorithme particulièrement efficace
- visualiseur graphique -> s'il est particulièrement bien foutu (vous êtes seul juge), vous pouvez compter deux graduations pour ce bonus-là.
- ...



Rate it from 0 (failed) through 5 (excellent)

Conclusion

Leave a comment on this correction

* (required) Comment

Finish correction

--