

[\(https://profile.intra.42.fr/\)](https://profile.intra.42.fr/)<https://profile.intra.42.fr/searches>

SCALE FOR PROJECT FT_CONTAINERS (HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/FT_CONTAINERS)

You should evaluate 1 student in this team

Git repository

`git@vogsphere-v2.42lyon.fr:vogsphere/intra-uuid-e1aec6ef-cb1b-495a-9c`

Introduction

Veuillez respecter les règles suivantes:

- Restez poli, courtois, respectueux et constructif pendant le processus d'évaluation. Le bien-être de la communauté repose là-dessus.
- Identifiez avec la personne (ou le groupe) évalué les éventuelles dysfonctionnements du travail. Prenez le temps d'en discuter et débitez des problèmes identifiés.
- Vous devez prendre en compte qu'il peut y avoir de légères différences d'interprétation entre les instructions du projet, son scope et ses fonctionnalités. Gardez un esprit ouvert et notez de la manière la plus honnête possible. La pédagogie n'est valide que si la peer-évaluation est faite sérieusement.

Disclaimer

- Ne notez que ce qui est contenu dans le dossier git cloné de l'étudiant ou du groupe.
- Vérifiez que le dossier git appartient bien à l'étudiant ou au groupe, que le projet correspond bien, et que "git clone" est utilisé sur un dossier vide.
- Vérifiez scrupuleusement qu'aucun alias n'a été utilisé pour vous tromper et assurez vous que vous corrigez bien le rendu officiel.
- Afin d'éviter toute surprise, vérifiez avec l'évalué les potentiels scripts utilisés pour faciliter l'évaluation.
- Si, en temps qu'évaluateur, vous n'avez pas fait le projet courant, vous devez lire le sujet en entier avant de commencer l'évaluation.
- Utilisez les flags disponibles pour signaler un rendu vide, un programme ne fonctionnant pas, une erreur de norme, un problème de triche... Dans ces situations, l'évaluation est terminée et la note est 0 (ou -42 en cas de triche). Cependant, à l'exception des cas de triche, vous êtes encouragés à continuer la discussion sur le travail rendu (même si pas terminé) pour identifier les problèmes qui ont causé cet échec et éviter de les reproduire dans le futur.
- Pendant toute la durée de l'évaluation, aucun segfault ou autre arrêt du programme (inattendu, prémature ou incontrôlé) ne sera toléré. Dans ces cas-là, la note finale est 0. Utilisez le flag approprié.
Vous ne devriez jamais avoir à éditer un fichier mis à part un fichier de configuration, si celui-ci existe. Dans le cas où vous souhaitez modifier un fichier, vous devez expliciter clairement les raisons de l'édition et être en accord avec l'étudiant évalué avant de faire

quoi que ce soit.

- Il vous revient également de vérifier l'absence de fuite mémoire. Toute mémoire allouée sur la heap doit être proprement libérée avant la fin de l'exécution.
Vous pouvez pour cela utiliser les différents outils présent sur l'ordinateur, tels que leaks, valgrind, ou encore e_fence. En cas de fuite mémoire, cochez le flag approprié.

Guidelines

Vous devez compiler avec clang++, with -Wall -Wextra -Werror -std=c++98
Nous vous rapellons que ce projet est en C++98, et toute fonction/container des versions suivantes n'est pas attendue,
et leur utilisation se soldera par un 0.

Ces points signifient que vous ne devez pas noter l'exercice demandé :

- Une fonction est implémentée dans un header (sauf pour les templates)
- Un Makefile compile sans les flags et/ou avec autre chose que clang++

Ces points signifient que vous devez tag le projet en "Fonction Interdite" :

- Utilisation d'une fonction "C" (*alloc, *printf, free)
- Utilisation d'une fonction interdite dans le projet
- Utilisation de "using namespace" ou de "friend"(dans ce sujet, «friend» est autorisé à certaines occasions).
- Utilisation d'une bibliothèque externe, ou de fonctionnalités C++20
- Utilisation d'un conteneur déjà existant, ou de toute fonction existante, pour implémenter un autre conteneur

Attachments

subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/41106/fr.subject.pdf>)

main.cpp (<https://projects.intra.42.fr/uploads/document/document/7032/main.cpp>)

Vérification des conteneurs

Verifiez que chaque container est bien implémenté

Vecteur basics

Assurez-vous que chaque fonction membre, overload et itérateur sont présents et fonctionnent comme prévu.
Utilisez le conteneur de bibliothèque standard pour vérifier que tout fonctionne de la même manière.

Yes

No

Vecteur Avancé

La structure de données interne doit être un tableau dynamique.
const_iterator et les itérateurs doivent être comparables.
Vérifiez le système de réallocation dynamique.
Testez la fonction swap.
Tous les itérateurs, pointeurs et références faisant référence à des éléments dans les deux conteneurs restent valides et
font désormais référence aux mêmes éléments auxquels ils faisaient référence avant l'appel, mais dans l'autre conteneur, où ils itèrent maintenant.
Vérifiez que le mot-clé friend est utilisé uniquement pour les opérateurs.

Yes

No

Performances vectorielles

Assurez-vous que la vitesse est raisonnable par rapport au vecteur STL !
Par exemple, une copie complète doit allouer toute la mémoire en un seul appel.

Yes

No

Map basics

Assurez-vous que chaque fonction membre, overload et itérateurs sont présents et fonctionnent comme prévu.
Utilisez le conteneur de bibliothèque standard pour vérifier que tout fonctionne de la même manière.

Yes

No

Map avancé

Vérifiez que la structure interne doit être un arbre ordonné (arbre AVL, arbre R-B...)
Vérifiez que la paire<> est recodée et utilisée.
ft::make_pair a fonctionné comme prévu.
Il n'y a jamais deux de la même clé dans une carte.
Vérifiez que le conteneur est commandé.
Vérifiez que std::allocator et allocator::rebind sont utilisés et qu'il n'y a pas d'utilisation directe de new.
Vérifiez que l'insertion ou la suppression n'invalident pas les itérateurs.
La fonction d'échange ne doit pas déplacer de données mais uniquement des pointeurs.
Le mot-clé Friend ne doit être utilisé que pour la surcharge d'opérateur.
Il n'y a pas de fuite de mémoire.

Yes

No

Performances de la Map

Assurez-vous que la vitesse est raisonnable par rapport à la STL !
Plus lent que la STL, c'est ok.
Un timeout complet ne l'est pas.
Si c'est plus de x20 plus lent que la STL, vous devriez compter faux.

Yes

No

Stack Basics

Assurez-vous que chaque fonction membre, overload et overload non-membre sont présentes et fonctionnent comme prévu.
Utilisez le conteneur de bibliothèque standard pour vérifier que tout fonctionne de la même manière.

Yes

No

Stack Advance

Les classes de conteneur standard vector, deque et list sont compatibles en tant que conteneur sous-jacent.
La stack ne peut pas être itérée.
Le conteneur sous-jacent doit être protégé et non privé !

Yes

No

Bonus

Si et seulement si la partie obligatoire est complète, vous pouvez continuer et noter les conteneurs bonus demandés dans le sujet.

Set

Assurez-vous que chaque fonction membre, overload et overload non-membre sont présentes et fonctionnent comme prévu.
Utilisez le conteneur de bibliothèque standard pour vérifier que tout fonctionne de la même manière.
La structure de données interne doit être un arbre noir et rouge.

Demandez les détails de l'étudiant.

Comment ça fonctionne ? si vous avez le moindre doute, ne comptez pas ce bonus, un `std::set` est plus simple qu'une carte.

Le seul bonus ici est pour l'arbre rouge et noir !

Yes

No

Ratings

Don't forget to check the flag corresponding to the defense

Ok

Outstanding project

Empty work

No author file

W Invalid compilation

Norme

Cheat

d Crash

Leaks

I Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

Privacy policy (<https://signin.intra.42.fr/legal/terms/5>)

Terms of use for video surveillance (<https://signin.intra.42.fr/legal/terms/1>)

Rules of procedure (<https://signin.intra.42.fr/legal/terms/4>)

Declaration on the use of cookies (<https://signin.intra.42.fr/legal/terms/2>)

General term of use of the site (<https://signin.intra.42.fr/legal/terms/6>)

Legal notices (<https://signin.intra.42.fr/legal/terms/3>)