Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the bene fits of the entire community.

SCALE FOR PROJECT FT_CONTAINERS (/PROJECTS/FT_CONTAINERS)

You should evaluate 1 student in this team

Git repository

Introduction

- Only grade the work that is in the student or group's GiT repository.

- Double-check that the GiT repository belongs to the student or the group. Ensure that the work is for the relevant project and also check that "git clone" is used in an empty folder.

- Check carefully that no malicious aliases were used to fool you and make you evaluate something other than the content of the official repository.

- To avoid any surprises, carefully check that both the evaluating and the evaluated students have reviewed the possible scripts used to facilitate the grading.

- If the evaluating student has not completed that particular project yet, it is mandatory for this student to read the entire subject prior to starting the defense.

- Use the flags available on this scale to signal an empty repository, non-functioning program, a norm error, cheating etc. In these cases, the grading is over and the final grade is 0 (or -42 in case of cheating). However, except for cheating, you are

encouragé à continuer à discuter de votre travail (même si vous ne l'avez pas terminé) pour identifier les problèmes qui ont pu causer cet échec et éviter de répéter la même erreur à l'avenir.

- Rappelez-vous que pendant toute la durée de la soutenance, pas de segfault, pas d'autre arrêt inattendu, prématuré, non contrôlé ou inattendu du programme, sinon la note finale est 0. Utilisez le drapeau approprié.

Vous ne devriez jamais avoir à modifier un fichier, sauf le fichier de configuration s'il existe. Si vous souhaitez modifier un fichier, prenez le temps d'en expliquer les raisons à l'étudiant évalué et assurez-vous que vous êtes tous deux d'accord.

- Vous devez également vérifier l'absence de fuites de mémoire. Toute mémoire allouée sur le tas doit être correctement libérée avant la fin de l'exécution.

Vous êtes autorisé à utiliser les différents outils disponibles sur l'ordinateur, tels que leaks, valgrind, ou e_fence. En cas de fuites de mémoire, cochez le drapeau approprié.

Avis de non-responsabilité

Veuillez respecter les règles suivantes :

- Restez poli, courtois, respectueux et constructif tout au long du processus d'évaluation. Le bien-être de la communauté en dépend.

- Identifiez-vous à la personne (ou au groupe) qui a évalué les éventuels dysfonctionnements du travail. Prenez le temps de discuter et de débattre des problèmes que vous avez identifiés.

- Vous devez tenir compte du fait qu'il peut y avoir une certaine différence dans la façon dont vos pairs ont compris les instructions du projet et l'étendue de ses fonctionnalités. Gardez toujours un esprit ouvert et notez-le le plus honnêtement possible. La pédagogie n'est valable que si et seulement si l'évaluation par les pairs est menée sérieusement.

Directives

Vous devez compiler avec clang++, avec -Wall -Wextra -
Werror Pour rappel, ce projet est en C++98.
Les fonctions membres ou les conteneurs C++11 (et ultérieurs) ne sont PAS attendus.

Dans tous les cas, vous ne devez pas noter l'exercice en question :
- Une fonction est implémentée dans un header (sauf dans un template)
- Un Makefile compile sans drapeaux et/ou avec autre chose que clang++.

Dans tous les cas, vous devez marquer le projet comme Fonction interdite :
- Utilisation d'une fonction "C" (*alloc, *printf, free)
- Utilisation d'une fonction non autorisée dans le sujet

- Utilisation de "using namespace" ou "friend" (dans ce sujet, "friend" est autorisé à des occasions spécifiques).
- Utilisation d'une bibliothèque externe, ou de fonctionnalités C++20
- Utilisation d'un conteneur déjà existant, ou de toute fonction existante, pour mettre en œuvre un autre conteneur.

Pièces jointes

Ill subject.pdf (https://cdn.intra.42.fr/pdf/pdf/26104/en.subject.pdf)

Ill main.cpp (/uploads/document/document/4164/main.cpp)

Contrôle des conteneurs

Vérifiez que chaque conteneur est correctement mis en œuvre.

Les bases du vecteur

Assurez-vous que chaque fonction membre, surcharge et itérateur est présent et fonctionne comme prévu. Utilisez le conteneur de la bibliothèque standard pour vérifier que tout fonctionne de la même manière.

Ill Oui Malade Non

Vector Advance

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

Ill Oui Malade Non

Performance vectorielle

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

Ill Oui Malade Non

Principes de base des cartes

Assurez-vous que chaque fonction membre, surcharge et surcharge non membre est présente et fonctionne comme prévu. Utilisez le conteneur de la bibliothèque standard pour vérifier que tout fonctionne de la même manière.

Ill Oui Malade Non

Avance de la carte

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

Ill Oui Malade Non

Performance de la carte

Assurez-vous que la vitesse est raisonnable par rapport à la STL !
Une vitesse inférieure à celle de la STL est acceptable.
Un temps d'arrêt complet ne l'est pas.
Si elle est plus de x20 plus lente que la STL, vous devez compter faux.

Ill Oui Malade Non

Les bases de l'empilage

Assurez-vous que chaque fonction membre, surcharge et surcharge non membre est présente et fonctionne comme prévu.
Utilisez le conteneur de la bibliothèque standard pour vérifier que tout fonctionne de la même manière.

Ill Oui Malade Non

Avance de la pile

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

Ill Oui Malade Non

Bonus

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

sujet.

Set

Make sure that each member function, overload and non-member overload are present and work as expected.
Use the standard library container to check that everything works the same way.
The inner data structure must be a Black and Red tree.
Ask the student details.
How it works ? if you have any doubt don't count this bonus a std::set is easier than a map.
The only bonus here is for Red and Black tree!

☑ Yes ☒ No

Ratings

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

✔ Ok ★ Outstanding project

Empty work | No author file | Invalid compilation | Norme | Cheat | Crash

Leaks | Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation