Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defenses during the Covid crisis allowed 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 benefits of the entire community.

SCALE FOR PROJECT PUSH_SWAP

You should evaluate 1 student in this team

Git repository

git@vogsphere-v2.42.fr:vogsphere/intra-uv1d-b834265f-6435-4c97-bc9b-

Introduction

Nous vous demandons pour le bon déroulement de cette évaluation de respecter les règles suivantes :

- Restez courtois, poli, respectueux et courtoisfs en toutes situations lors de cet échange. Le lien de confiance entre la communauté 42 et vous en dépend.

- Mettez en évidence auprès de la personne (ou du groupe) notre les dysfonctionnements éventuels du travail rendu, et prenez le temps d'en discuter et d'en débattre.

- Acceptez 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 (s'il ou elle raison ou tort !), et notez le plus honnêtement possible. La pédagogie de 42 n'a de sens que si la peer-évaluation est faite sérieusement.

Guidelines

- Vous ne devez évaluer que ce qui se trouve sur le dépôt GIT de rendu de l'étudiant(e) ou du groupe.

- Prenez soin de vérifier que le dépôt GIT est bien celui correspondant à l'étudiant(e) ou au groupe, et au projet, et que "git clone" s'opère bien utilisé dans un répertoire vide.

- Vérifiez méticuleusement qu'aucun alias malicieux n'a été utilisé pour nous induire en erreur et vous faire évaluer autre chose que le contenu du dépot officiel.

- Tout script serait facilitant l'évaluation fourni par l'un des deux parts doit être rigoureusement vérifié pour l'autre partt pour éviter des mauvaises surprises.

- Si l'étudiant(e) corrective/correctrice n'a pas encore fait ce projet, il est obligatoire pour ce(tte) etudiant(e) de lire le sujet en entier avant de commencer cette soutenance.

- Utilisez les flags disponibles sur ce barème pour signaler un rendu vide, non fonctionnel, une faute de norme, un cas de triche, etc. Dans ce cas, l'evaluation est terminée et la note finale est 0 (ou -42 dans le cas spécial de la triche). Toutefois, hors cas de triche, vous etes encouragés à continuer d'échanger autour du travail effectué (ou non effectué justement) pour identifier les problèmes ayant entraîné cette situation et les éviter pour le prochain rendu.

Attachments

subject.pdf checker_Mac checker_linux

Partie obligatoire

Rappel : si a un moment ou un autre, le programme ne réagit pas correctement (bus error, segfault, etc...), ou bien si vous détectez une fuite mémoire, la soutenance est terminée et la note est 0. Pensez à utiliser les flags correspondants quand nécessaire. Cette consigne est active d'un bout à l'autre de la soutenance.

Fuites mémoire

Pendant toute la durée de la soutenance, gardez un oeil sur la quantité de mémoire utilisée par le push_swap (à l'aide de top par exemple) pour détecter des anomalies, et vérifiez dans le code que les allocations sont systématiquement libérées. Dans le cas contraire il y a eu moins une fuite mémoire, la note du projet est 0 :

Yes No

Gestion des erreurs

Dans cette section nous allons évaluer la bonne gestion des erreurs de votre push_swap.
Si au moins l'un des tests échoue, aucun point n'est gagné pour cette question. Passez à la question suivante.

- Lancer push_swap avec des paramètres non numériques. Le programme doit afficher "Error".

- Lancer push_swap avec un doublon dans les paramètres. Le programme doit afficher "Error".

- Lancer push_swap avec seulement des nombres en paramètre, l'un d'entre eux étant supérieur à MAXINT. Le programme doit afficher "Error".

- Lancer push_swap avec aucun parametre. Le programme ne doit rien afficher, se terminer, et le shell affiche son prompt.

Yes No

Push_swap - Identité

Nous allons évaluer dans cette section le comportement du programme push_swap sur une liste de parametres déjà triés. Effectuez les 3 tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez "$./push_swap 42". Le programme affiché doit être vide (0 instruction).

- Lancez "$./push_swap 0 1 2 3". Le programme affiché doit être vide (0 instruction).

- Lancez "$./push_swap 0 1 2 3 4 5 6 7 8 9". Le programme affiché doit être vide (0 instruction).

Yes No

Push_swap - tests simples

Si le test suivant échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez "$ARG="2 1 0", ./push_swap $ARG | ./checker_OS $ARG". Verifiez que le programme checker affiche OK que la taille du programme calculé par le programme push_swap est de 2 OU 3 instructions. Sinon le test est échoué.

Yes No

D'autres tests tout aussi simples

Si l'un des 2 tests suivants échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez "$ARG="1 5 2 4 3", ./push_swap $ARG | ./checker_OS $ARG". Verifiez que le programme checker affiche OK et que la taille du programme calculé par le programme push_swap est de 12 instructions au maximum. Sinon le test est échoué. Kudos pour la personne évaluée si la taille programme est de 8 instructions.

- Lancez "$ARG="<5 params au choix>", ./push_swap $ARG | ./checker_OS $ARG" en remplaçant le placeholder par 5 valeurs valides de votre choix. Verifiez que le programme checker affiche OK et que la taille du programme calculé par le programme push_swap est de 12 instructions au maximum. Sinon le test est échoué. Vous devez en particulier vérifier avec ce test que le programme push_swap n'a pas été programmé pour répondre correctement uniquement aux tests de ce barème. Vous êtes encouragés à répéter ce test avec plusieurs variantes avant de le valider.

Yes No

Push_swap - tests intermédiaires

Si le test suivant échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez "$ARG="<100 params au choix>", ./push_swap $ARG | ./checker_OS $ARG" en remplaçant le placeholder par 100 valeurs valides de votre choix (un vous appelle pas Jeanfred-Michel&Co). Script pour ceci. Verifiez que le programme checker affiche OK et mettez des points en fonction du nombre d'instructions effectuées
- Moins de 700: 5
- Moins de 900: 4
- Moins de 1100: 3
- Moins de 1500: 2
- Moins de 1000: 1

Vous devez en particulier vérifier avec ce test que le programme push_swap n'a pas été programmé pour répondre correctement uniquement aux tests de ce barème. Vous êtes encouragés à répéter ce test avec plusieurs variantes avant de le valider.

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

4

Push_swap - tests avancés

Si le test suivant échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez "$ARG="<500 params au choix>", ./push_swap $ARG | ./checker_OS $ARG" en remplaçant le placeholder par 500 valeurs valides de votre choix (un vous appelle pas Jeanfred-Michel&Co). Script pour ceci. Verifiez que le programme checker affiche OK et mettez des points en fonction du nombre d'instructions effectuées
- Moins de 5500: 5
- Moins de 7000: 4
- Moins de 8500: 3
- Moins de 10000: 2
- Moins de 11500: 1

Vous devez en particulier vérifier avec ce test que le programme push_swap n'a pas été programmé pour répondre correctement uniquement aux tests de ce barème. Vous êtes encouragés à répéter ce test avec plusieurs variantes avant de le valider.

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

3

Bonus

Rappel : si a un moment ou un autre, le programme ne réagit pas correctement (bus error, segfault, etc...), la soutenance est terminée et la note est 0. Pensez à utiliser les flags correspondants. Cette consigne est active d'un bout à l'autre de la soutenance. Les bonus ne doivent être évalués que si et seulement si la partie obligatoire est PARFAITE. Par PARFAITE, on entend bien évidemment qu'elle est entièrement réalisée, qu'il n'est pas possible de mettre son comportement en défaut, même en cas d'erreur, aussi vicieuse soit-elle, de mauvaise utilisation, etc. Concrètement, cela signifie que si la partie obligatoire n'a pas obtenu TOUS les points pendant cette soutenance, les bonus doivent être intégralement IGNORÉS.

Checker - gestion des erreurs

Nous allons évaluer dans cette section la gestion d'erreur du programme checker. Effectuez les tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez checker avec des parametres non numériques. Le programme doit afficher "Error".

- Lancez checker avec 2 fois le même paramètre numérique. Le programme doit afficher "Error".

- Lancez checker avec uniquement des parametres numériques dont l'un plus grand que MAXINT. Le programme doit afficher "Error".

- Lancez checker sans aucun parametre. Le programme doit rendre la main et ne rien afficher.

- Lancez checker avec des parametres valides, puis au moment d'entrer les instructions à exécuter, entrez des instructions qui n'existent pas. Le programme doit afficher "Error".

- Lancez checker avec des parametres valides, puis au moment d'entrer les instructions à exécuter, entrez des instructions valides, mais précédées les et terminez les avec un ou plusieurs espaces. Le programme doit afficher "Error".

Yes No

Checker - tests des faux résultats

Nous allons évaluer dans cette section le comportement du programme checker lorsque le programme lu ne trie pas la liste. Effectuez les 2 tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

N'oubliez pas d'appuyer sur ctrl+d pour arrêter la lecture une fois que vous avez fini d'entrer les instructions.

- Lancez checker avec la commande "$./checker 0 9 1 8 2 7 3 6 4 5" puis entrez les instructions suivantes avec un formatage valide : [sa, pb, rra]. Le programme checker doit afficher "KO".

- Lancez checker avec une liste de paramètres valides de votre choix, puis entrez les instructions de votre choix avec un formatage valide, mais qui ne permettent pas de trier les entiers. Le programme checker doit afficher "KO". Vous devez en particulier vérifier avec ce test que le programme checker n'a pas été programmé pour répondre correctement uniquement aux tests de ce barème. Vous êtes encouragés à répéter ce test avec plusieurs variantes avant de le valider.

Yes No

Checker - test des résultats corrects

Nous allons évaluer dans cette section le comportement du programme checker lorsque le programme lu trie la liste. Effectuez les tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

N'oubliez pas d'appuyer sur ctrl+d pour arrêter la lecture une fois que vous avez fini d'entrer les instructions.

- Lancez checker avec la commande "$./checker 0 1" puis appuyez sur ctrl+d sans entrer d'instructions. Le programme checker doit afficher "OK".

- Lancez checker avec la commande "$./checker 0 9 1 8 2" puis entrez les instructions suivantes avec un formatage valide : [pb, ra, pb, ra, sa, ra, pa, pa]. Le programme checker doit afficher "OK".

- Lancez checker avec une liste de paramètres valides de votre choix, puis entrez les instructions de votre choix avec un formatage valide qui permettent de trier les entiers. Le programme checker doit afficher "OK". Vous devez en particulier vérifier avec ce test que le programme checker n'a pas été programmé pour répondre correctement uniquement aux tests de ce barème. Vous êtes encouragés à répéter ce test avec plusieurs variantes avant de le valider.

Yes No

Ratings

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

Ok Outstanding project

Empty work Incomplete work No author file Invalid compilation

Norme Cheat Crash Leaks Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation