

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 au moins une fuite mémoire, la note du projet est 0.

Programme checker – Gestion des erreurs

Nous allons évaluer dans cette section la gestion d'erreur du programme checker. Effectuez les 4 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 paramètres 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 paramètres numériques dont l'un plus grand que MAXINT. Le programme doit afficher "Error".
- Lancez checker sans aucun paramètre. Le programme doit rendre la main et ne rien afficher.
- Lancez checker avec des paramètres 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 paramètres valides, puis au moment d'entrer les instructions à exécuter, entrez des instructions valides, mais précédez les et terminez les avec un ou plusieurs espaces. Le programme doit afficher "Error".

Programme checker – Tests faux

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, rrr]. Le programme checker doit afficher "K0".

– 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.

Programme checker – Tests vrais

Nous allons évaluer dans cette section le comportement du programme checker lorsque le programme lu trie 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 1 2" 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.

Push_swap – Test identité

Nous allons évaluer dans cette section le comportement du programme push_swap sur une liste de paramètres 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 instructions).

– Lancez "\$>./push_swap 0 1 2 3". Le programme affiché doit être vide (0 instructions).

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

Push_swap – Version simple

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 \$ARG". Vérifiez 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.

Version toute aussi simple

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 \$ARG". Vérifiez 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 \$ARG" en remplaçant le placeholder par 5 valeurs valides de votre choix. Vérifiez 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.

Push_swap – Version Intermédiaire

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 \$ARG" en remplaçant le placeholder par 100 valeurs valides de votre choix. Vérifiez 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 1300: 2
 - Moins de 1500: 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.

Push_swap – Version avancée

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 \$ARG" en remplaçant le placeholder par 500 valeurs valides de votre choix (on vous appelle pas Jean(ne)-Michel(le) Script pour rien). Vérifiez 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.