

Exercice 4

Difficulté : 63 points-virgules

On désire réaliser un programme pour calculer le produit scalaire de deux vecteurs $X = (x_1, x_2 \dots x_n)$ et $Y = (y_1, y_2 \dots y_n)$, où $n \geq 1$ et les x_i et y_i sont des entiers pour simplifier. Comme faire ce calcul directement serait trop simple, on utilisera $c + 1$ nouveaux processus et deux tubes.

Votre programme doit admettre $2n + 1$ arguments :

```
prodscal c x1 x2 ... xn y1 y2 ... yn
```

Pour calculer ce produit scalaire, on utilisera la méthode suivante :

1. le processus père crée deux tubes ;
2. le processus père génère $c + 1$ processus fils p_j ($j \leq c$) ;
3. le processus père écrit ensuite les différents couples (x_i, y_i) dans le premier tube (avec $1 \leq i \leq n$) ;
4. chaque processus p_j (avec $0 \leq j < c$) lit de manière répétitive des couples (x_i, y_i) dans le premier tube et écrit dans le deuxième tube le produit de chaque couple ; il s'arrête lorsqu'il n'y a plus de donnée à lire dans le tube ;
5. le processus p_c lit les produits dans le deuxième tube et en calcule la somme. Il affiche le résultat et s'arrête lorsqu'il n'y a plus de donnée à lire dans le deuxième tube ;
6. enfin, le processus père attend tous les fils, et renvoie un code de retour nul s'ils se sont tous terminés sans erreur.

Par exemple :

```
> ./prodscal 2 1 3 5 7 -9 8 7 6
92
```

Pour rédiger votre programme, il est impératif de respecter les contraintes suivantes :

- vous ne devez utiliser que les primitives système (ou assimilées comme telles) ; vous pouvez toutefois utiliser les fonctions de bibliothèque pour les affichages ;
- pour des raisons d'efficacité, vous ne ferez pas d'appels redondants à des fonctions lentes (primitives système ou autres) ;
- lorsqu'une erreur est détectée, le programme doit s'arrêter aussitôt avec un code de retour indiquant l'erreur, sans attendre la terminaison de tous les processus fils en cours d'exécution ;
- votre programme doit retourner un code de retour nul (`exit(0)`) si tout s'est déroulé sans erreur ou un code de retour non nul (`exit(1)`) si une erreur a été rencontrée ;
- si votre programme est appliqué avec un nombre d'arguments incorrect, il doit afficher le message : `"usage: prodscal c x1 ... xn y1 ... yn"`.
- vous apporterez un soin particulier à la mise en forme de façon à rendre un code lisible et commenté à bon escient. Référez-vous au document « Conseils pour réussir vos TP et projets » mis à votre disposition sur Moodle et, si besoin, utilisez l'utilitaire `clang-format` avec la configuration donnée dans ce document ;
- votre programme doit compiler avec les options `-Wall -Wextra -Werror -pedantic` sur `gcc` version 9.4 minimum (la version disponible sur la machine `turing.u-strasbg.fr`). Alternativement, vous pouvez utiliser l'image Docker `pdagog/refc` (version de `gcc` 13.2) Les programmes qui ne compilent pas au moins sur `turing` avec ces spécifications **ne seront pas examinés**.

Un script de test est mis à votre disposition sur Moodle. Celui-ci exécute votre programme sur des jeux de tests qui serviront de base à l'évaluation de votre rendu. La commande suivante permet de lancer les tests : `sh test4.sh`.

Vous devrez rendre sur Moodle un *unique* fichier nommé `prodscal.c`.

Cet exercice est **individuel**. On rappelle que la copie ou le plagiat sont sévèrement sanctionnés.