

Programmation - Projet tableaux

3 octobre 2017

1 Consignes

1. Une fois votre projet achevé, vous devez rendre votre fichier C sur “syntax error”.
2. La date de fin du projet est celle fixée sur “syntax error”. Aucune soumission après cette date ne sera acceptée.
3. Il est formellement interdit de recopier des lignes de code d’une tierce personne. Toute tricherie avérée entraînera la note de zéro (l’entraide reste bien sûr autorisée et encouragée).

Restrictions :

1. Les seules fonctions de la libC autorisées sont `printf` et `scanf`.

2 Le travail

Vous devez faire un programme dans un fichier `calcul.c` permettant de faire des opérations arithmétiques sur des grands nombres. Votre programme devra lire sur l’entrée standard quelque chose du type :

$$n_1 \square n_2 \triangle n_3$$

où :

- n_1 , n_2 et n_3 sont des nombres entiers pouvant comporter jusqu’à 100 chiffres.
- Les nombres n_1 , n_2 et n_3 peuvent être négatifs, auquel cas ils commencent par le signe ‘-’.
- \square et \triangle sont tous les deux un des opérateurs suivant :
 - ‘-’ pour la soustraction
 - ‘+’ pour l’addition
 - ‘*’ pour la multiplication
 - ‘/’ pour la division entière
 - ‘%’ pour le reste de la division entière
- Il peut y avoir autant d’espaces que l’on veut (y compris aucun espace) entre les nombres et les différents opérateurs (y compris le moins unaire, indiquant qu’un nombre est négatif).
- Le deuxième opérateur \triangle avec l’opérande n_3 sont facultatifs. Ils peuvent apparaître ou peuvent ne pas apparaître.
- Votre programme s’arrête de lire quand il arrive au caractère ‘\n’.

Une fois que votre programme a lu le calcul à faire (cette lecture se termine quand on atteint le caractère ‘\n’), il doit afficher le résultat de ce calcul sur la sortie standard, suivit d’un retour à la ligne, puis il doit s’arrêter. De plus votre programme doit tenir compte de la priorité des opérateurs *, / et % sur les opérateurs + et -. On pourra supposer que l’entrée est toujours correcte (et en particulier qu’il n’y a pas de divisions par zéro).

Vous pouvez utiliser le programme `bc` pour tester votre projet. La commande `echo "3+4*2" | bc` affiche le résultat du calcul `"3+4*2"`, tout comme doit le faire votre programme. La variable d'environnement `BC_LINE_LENGTH` indique la longueur des lignes affichées par `bc`. Par défaut `BC_LINE_LENGTH` a une valeur de 100, ce qui veut dire qu'un nombre de 250 chiffres sera affiché sur 3 lignes. Contrairement à `bc`, Votre programme ne doit bien sûr jamais revenir à la ligne avant d'avoir affiché tout le résultat.

```
$> gcc -o calcul calcul.c
$> echo " 234 + - 22 * 58 " | ./calcul | cat -e
-1042$
$> echo "100374647382947463*28476463839746376382-163748" | ./calcul | cat -e
2858315016627796681514500552929855118$
$>
```

Il y a plusieurs manière de considérer le modulo sur les nombres négatifs. On utilise ici la même convention que `bc` : $-a \% b = -(a \% b)$ et $a \% -b = a \% b$.

Conseils :

1. Ne paniquez pas.
2. Une méthode possible :
 - (a) Stockez chaque entier dans un tableau de taille 100 (chaque case doit contenir un chiffre de l'entier).
 - (b) Déterminez la taille maximale que doit avoir un tableau pour contenir le résultat (un tableau de taille 1000 devrait suffir quoi qu'il arrive).
 - (c) Pour chaque opération, ayez une fonction, qui appliquera l'algorithme correspondant, chiffre par chiffre, comme vous faisiez à l'école primaire, et stockera le résultat dans un troisième paramètre (tableau de taille 1000).