

Cours "Informatique Embarquée"
François Armand
M2 IAP/Crypto/EIDD
Exercice N° 1, 2 Octobre 2015
A rendre avant le 9 Octobre 23H59 sur Didel
2 séances de TP.
armand@informatique.univ-paris-diderot.fr

Rappels

- Voir la note sur le contrôle d'Informatique Embarquée
- Voir la note sur les TPS

Dépôt intermédiaire !

Il vous est proposé, pour ce TP, de déposer votre code/TP « en l'état » à la fin de la première séance de TP (avant 19H). Ce travail intermédiaire sera rapidement examiné et des commentaires vous seront envoyés, aussi vite que possible via Didel.

Sujet du TP N°1

Avant de vous lancer dans la programmation, vous tenterez d'estimer le temps nécessaire à la réalisation (réflexion, codage, tests, écriture du rapport) de ce TP.

Le travail demandé est de remplir un tableau à deux dimensions de tailles quelconques (M, N) en numérotant les cases de manière croissante en colimaçon (hélice ou spirale).

- M et N sont supérieurs ou égaux à 1
- M et N peuvent être **très grands**
- La valeur initiale est 1

Le langage de programmation est impérativement le C. Vous structurerez votre code de manière à fournir une fonction ayant **OBLIGATOIREMENT** le prototype suivant :

```
int colimacon(int **array, unsigned int rows,
              unsigned int columns)
```

array : adresse d'un pointeur sur entier. Au retour de la fonction, array contiendra l'adresse du tableau qui aura été alloué et « rempli » par la fonction colimacon, en cas de succès et NULL sinon.

rows : valeur fournie par l'appelant définissant le nombre de lignes du tableau.

columns : valeur fournie par l'appelant définissant le nombre de colonnes du tableau.

La fonction colimacon procédera à l'allocation mémoire du tableau et le remplira. En cas de succès, la fonction renverra la valeur 1, et 0 en cas d'échec.

Réfléchissez bien ! Le prototype de la fonction ci-dessus contraint fortement votre implémentation.

Le remplissage doit se faire comme dans l'exemple ci-dessous :

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

Votre fonction `colimacon` sera fournie sous forme de bibliothèque (.a ou .so à votre convenance). Vous fournirez aussi un programme « main » invoquant cette fonction et permettant de tester le comportement de cette fonction selon les arguments passés au programme et indiquant le nombre de lignes et de colonnes souhaitées. Ce programme imprimera le tableau construit par la fonction `colimacon`. Ex, la commande suivante imprimera un tableau de 3 lignes et 4 colonnes :

```
# colimacon 3 4
```

Vous fournirez aussi un programme de test automatique validant votre fonction `colimacon` sur un certain nombre d'exemples.

Dans votre compte-rendu, vous expliquerez:

- Vos choix,
- Les problèmes de mise au point que vous avez rencontrés,
- Quel prototype préféreriez-vous pour la fonction « colimacon » ?
- Quel impact votre algorithme a vis à vis des accès mémoire. (Le programme doit pouvoir fonctionner sur des tableaux de 10 000 x 10 000 ou plus encore.)
- Votre programme pourrait-il être parallélisé, et fonctionner dans un programme multi-threads ? Sinon, serait-il possible de concevoir un algorithme permettant facilement une parallélisation, argumentez.
- Vous consignerez l'estimation initiale que vous avez faite du temps de travail nécessaire.
- Vous consignerez aussi le temps que vous avez réellement passé pour le codage et la mise au point de ce programme.
- Vous rapporterez les commentaires faits par le groupe / les étudiants à qui vous aurez soumis votre travail pour relecture. Vous direz quelles remarques vous avez prises en compte. Vous justifierez votre rejet des remarques non prises en compte.
- Questions subsidiaires :
 - Quel(s) est (sont) les temps nécessaire(s) à l'exécution de votre programme sur des tableaux de 10x10, 100x100, 1000x1000 et 10 000 x 10 000.
 - Quelle est l'occupation mémoire de votre programme dans les cas ci-dessus ?
 - Vous fournirez éventuellement une solution alternative à votre solution.