

Cours " Informatique Embarquée "

François Armand

M2 IMPAIRS/Crypto/EI

E!er"i"e # \$ % & ' ("tobre 2)*

A rendre a+ant ,e %- ("tobre 2* .

/0imite impérati+e1 sur ide,

armand 2 informatique3uni+4paris4diderot3fr

Rappe,s

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

Dépôt intermédiaire bonifié!

Il vous est proposé, titre e!périmental pour ce TP, de déposer votre code"TP # en l'état \$ la fin de la premi%re séance de TP. 'e travail intermédiaire sera e!aminé et des commentaires vous seront envo(és, aussi vite que possible.)e fait de déposer votre travail en fin de séance de TP *dans un délai ma!imum d'une demi+,eure apr% la fin du TP- , devrait'-' pris en considération *favorablement- dans la note de TP attribuée sur le # rendu \$ final.

*-' devrait' 'onditionnel. Il s'a/it d'un essai.

Su0et du TP 1 23

A+ant de +ous ,an"er dans ,a pro5rammation. +ous tentere6 d'estimer ,e temps né"essaire 8 ,a réa,isation /réf,e!ion. "oda5e. tests. é"riture du rapport1 de "e 9P.

)e travail demandé est de remplir un tableau deu! dimensions de tailles quelconques * 4 , 1 - en numérotant les cases de mani%re croissante en colima5on *,élice ou spirale-.

- 4 et 1 sont supérieurs ou é/au! 3
- 4 et 1 peuvent être tr:s 5rands
-)a valeur initiale est 3

)e lan/a/e de pro/rammation est impérativement le ' & Vous structurere7 votre code de mani%re fournir une fonction a(ant (; 01 < A9 (IREME#9 le protot(pe suivant :

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

a r r a y & adresse d'un pointeur sur entier. Au retour de ,a fon"tion. array "ontindra ,=adresse du tableau qui aura été a,,oué et > remp,i ?par ,a fon"tion co l i m a c o n. en cas de succ% et NULL sinon.

r o w s & valeur fournie par l'appelant définissant le nombre de li/nes du tableau.

c o l u m n s & valeur fournie par l'appelant définissant le nombre de colonnes du tableau.

La fonction collimacron produira 8, la notation mémoire du tableau et le remplira. En cas de succès, la fonction renverra la valeur 3, et 9 en cas d'écoulement.

; réfléchissez bien !) le prototype de la fonction ci-dessus contraint fortement votre implémentation.

le remplissage doit se faire comme dans l'exemple ci-dessous.

Exemple

3	<	=	>
3<	3=	3>	?
33	3@	3?	@
39	A	B	C

Votre fonction `colimacon` sera fournie sous forme de bibliothèque, que ce soit 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. Le programme imprimera le tableau construit par la fonction `colimacon`.

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

Dans votre compte-rendu, vous expliquerez :

- Vos conclusions,
- Les problèmes de mise au point que vous avez rencontrés,
- L'impact votre algorithme a eu vis-à-vis des accès mémoire. Votre programme doit pouvoir fonctionner sur des tableaux de 39 999 à 39 999 ou plus encore.
- Votre programme pourrait-il être parallélisé, et fonctionner dans un programme multi-threadés. Sinon, serait-il possible de concevoir un algorithme permettant facilement une parallélisation, argumentée.
- Vous consignez l'estimation initiale que vous avez faite du temps de travail nécessaire.
- Vous consignez aussi le temps que vous avez réellement passé pour le coder et la mise au point de ce programme.
- Vous rapporterez les commentaires faits par le jury. Les étudiants qui vous auront 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 est le temps nécessaire à l'exécution de votre programme sur des tableaux de 39 à 39, 399 à 399, 3999 à 3999 et 39 999 à 39 999.
 - Quelle est l'occupation mémoire de votre programme dans les cas ci-dessus.
 - Vous fournirez éventuellement une solution alternative à votre solution.

Autres suggestions / propositions

Les exercices qui suivent ne donneront pas lieu à notation, mais seront corrigés si vous les réalisez.

3. Procéder de manière identique pour une fonction palindrome qui détermine si une chaîne de

caractères est un palindrome

```
int palindrome (char * s)
```

< De la même manière, on pourra implémenter une fonction

```
int check_moves_in_array(int *array, unsigned int size)
```

'ette fonction reçoit un tableau d'entiers initialisés `array` qui comporte `size` entrées

'ette fonction lit initialement le contenu de :

- `x=array[0]`
- `(Garray[0 + x]`
- `7 Garray[x + y]`
- `73 Garray[y + z]`
- etc

Evidemment en fonction du contenu des cellules, on peut être amené :

- sortir du tableau
- boucler indéfiniment dans le tableau

Dans le premier cas *sortir du tableau- la fonction renverra le dernier indice accédé dans le tableau. Dans le même cas *boucle infinie-, on sortira de la fonction avec la valeur +3