

TP n° 1**Consignes pour le rendu**

Relire cette feuille avant d'envoyer le mail !

Date limite d'envoi : **dimanche 5 octobre 2014 à 23h59.**

Le rendu se fera sous forme d'une archive (zip ou tar), contenant :

1. Un **unique fichier de rendu au format PDF** comprenant les réponses à toutes les questions sauf les implémentations en C de la question (c), et les script Bash (ou autres) de la question (d).
2. Une série de quatre programmes nommées **max1.c**, **max2.c**, **max3.c**, **max4.c** correspondants aux quatre points de l'énoncé. Je vais compiler chacun avec une commande du type

```
gcc max1.c -o max1.
```

3. Si vous choisissez Bash (sinon faites quelque chose de similaire) pour la question (d) : un ou quatre scripts Bash, nommé **test.sh**, ou **test1.sh**, **test2.sh**, **test3.sh**, **test4.sh**.

Envoyer l'archive à l'adresse **kevin.perrot@lif.univ-mrs.fr**, avec comme intitulé :

[Complexité] Rendu du TP1 : <nom1>, <nom2>.

Remplacer <nom1> et <nom2> par vos noms, ou par un seul nom si vous rendez le TP seul.

De plus :

- (a) Avant le pseudo-algorithme, décrire en quelques phrases son fonctionnement.
- (b) Expliquer en quelques phrases vos résultats de complexité dans le pire cas.
- (c) Convention à respecter **IMPÉRATIVEMENT** :
 - Les noms des fichiers indiqués plus haut ;
 - Vos programmes C doivent prendre en argument une liste d'entiers relatifs comme suit :

```
./max1 -2 4 5 -4 5 6 9
```

et écrire à l'écran le résultat, par exemple ici : « 1 6 »
(**IMPORTANT** : la première case du tableau a pour indice 0).

(d) Pour répondre à cette question le style est libre, il faut faire comme vous êtes le plus à l'aise, expliquer simplement en quelques phrases vos choix dans le rendu.

Je vous donne les informations suivantes, qui pourront par exemple vous aider à répondre à l'aide d'un script `Bash` :

- Pour mesurer le temps d'exécution d'un programme, vous pouvez utiliser la fonction `time` en `Bash`. Son fonctionnement est simple, pour mesurer le temps d'exécution de l'exécutable `mon_programme` sur l'entrée `2 -3 6`, on fait

```
time ./mon_programme 2 -3 6.
```

Consulter `man time` pour comprendre le résultat.

- Pour générer des instances aléatoires, vous pouvez utiliser la variable `RANDOM` en `Bash`. Elle change aléatoirement de valeur à chaque consultation. Vous pouvez partir de

```
echo $(( $RANDOM % 100 ))
```

qui écrit un nombre aléatoire entre 0 et 99.

(e) Rien à signaler.