

# TP2(bis) C++ – M2 ISIFAR

Exercices supplémentaires sur les pointeurs

M. de Falco [defalco@pps.jussieu.fr] – Site de Chevaleret, Bureau 5A16

13 octobre 2009

## 1 Les références, les pointeurs et les paramètres des fonctions

### 1.1 Question 1

On définit trois nombres réels  $a$ ,  $b$ ,  $c$  simple précision (float) dans le `main()`, et on leur attribue des valeurs arbitraires, par exemple  $a=11.5$ ,  $b=-2.1$  et  $c=0.0$ . On désire écrire une fonction `ordonne_trois()` qui puisse manipuler ces trois variables passées en paramètre d'une certaine façon, et faire en sorte qu'après son appel, on ait toujours la situation  $a \leq b \leq c$ , éventuellement en permutant leurs valeurs.

Afin de simplifier le problème, on décide d'écrire d'abord une fonction `ech2_par_adresse()` qui effectue le tri sur seulement deux paramètres à la fois. Ecrivez cette fonction puis testez là sur deux variables du `main()`.

Une fois que `ech2_par_adresse()` est au point, écrivez `ordonne_trois()`. de telle façon qu'elle fasse un certain nombre d'appels à `ech2_par_adresse()`. Combien faut-il d'appel (au minimum) à `ech2_par_adresse()` pour être certain de la justesse du résultat ?

Pour être certain que votre solution soit juste il faut que vous ne fassiez les affichages que dans le `main()`. Ce qui interdit en particulier d'en faire dans `ech2_par_adresse()` ou dans `ordonne_trois()`.

### 1.2 Question 2

On reprend la question précédente mais on décide d'utiliser des passages par référence à la place des passages par adresse. Laquelle des deux solutions vous paraît la plus simple ?

## 2 Fonction en paramètre

Ecrire une fonction `maximum` qui prend en entrée deux entiers et une fonction de comparaison entre entiers et renvoie le plus grand des deux en l'utilisant.

## 3 Retour de références

Que se passe-t-il si votre fonction renvoie une référence ?

On étudiera cette question en écrivant de courts exemples où la référence est issu d'un argument en référence, d'une variable locale, d'une valeur passée par adresse.

Si on a une fonction

```
int & identite(int &x) { return x; }
```

que se passe-t'il en cas d'appel à `identite(x) = 2; ?`