PARTIEL

License maths-info : Informatique pour les scientifiques
mercredi 17 février 2010

Lisez attentivement ces instructions avant de commencer :

- Lisez tout l'énoncé avant de commencer. Ne restez pas bloqué sur un exercice, ils sont conçus pour être dans une certaine mesure indépendants.
- Le barême est donné à titre indicatif, mais il est possible d'avoir 20 sans tout faire.
- Bon courage.

Partie Écrite

1. (*1 point*) Donnez le prototype C des fonctions suivantes :

(**a**) $f : \mathbb{R} \to \mathbb{R}$
(**b**) $g : \mathbb{Z} \times \mathbb{R} \to \mathbb{R}$

2. (*1 point*) Donnez le type des prototypes suivants :

(**a**) `unsigned int h(int,double,int);`
(**b**) `double k(double,double);`

3. (*1 point*) Que va-t-il se passer si j'exécute trop souvent la fonction suivante (`compteur` est une variable globale de type `int`) ?

```
void incremente() {
  compteur = compteur+1;
  printf("le compteur vaut %i\n",compteur);
}
```

4. (*3 points*) Écrivez une fonction C qui, étant donné le numéro d'un mois (un entier entre 1 et 12), renvoie le booléen `VRAI` s'il a 31 jours, `FAUX` sinon.

5. (*4 points*) Voyez-vous un problème avec les codes suivants (la fonction $sqrt : \mathbb{R} \to \mathbb{R}$ calcule la racine carrée et $log : \mathbb{R} \to \mathbb{R}$ le logarithme en base 10. Elles sont fournies par `math.h`) ? Si oui, quelle(s) correction(s) proposz-vous ?

(**a**)

```
#include <math.h>

double un_sur_sqrt(double x) {
  return 1.0/sqrt(x);
}
```

(**b**)

```
#include <math.h>

booleen log_is_positive(double x) {
  if(log(x)>0 && x>0) {
    return VRAI;
  }

  return FAUX;
}
```

Partie sur Machine

6. (*2 points*) Nous allons créer une bibliothèque `bibpartiel` (qui ne sera utilisée que pour ce partiel). Créez un répertoire `partiel`. Tout va se passer dans ce répertoire. Vous en ferez une archive que vous déposerez sur arche à la fin de la séance. Tout retard entraînera une pénalité. Copiez-y `biblocale`. Créez-y un répertoire `bibpartiel`. Dans ce répertoire vous écrirez :

- le fichier `bibpartiel.h`
- le fichier `bibpartiel.c`
- le `Makefile` pour compiler la bibliothèque

7. (*1 point*) Dans le répertoire `partiel`, créez :
- un fichier `main.c` qui utilisera `biblocale` et `bibpartiel` (vous devez donc `inclure` les bons headers). Pour l'instant la fonction `main` est vide (elle se contente de renvoyer 0).
- un `Makefile` pour compiler votre main.

8. (*1 point*) Dans la bibliothèque, créez une fonction `affiche_10` qui affiche les 10 premiers entiers (0,1,2,..,9). Testez-la. Quel est son type ?

9. (*2 points*) Dans la bibliothèque, créez une fonction `affiche` qui prend deux entiers A et B avec $A \leq B$ et affiche tous les entiers entre ces deux là. Testez-la (y compris la robustesse). Quel est son type ?

10. (*3 points*) Dans la bibliothèque, créez une fonction `somme` qui lit des entiers sur l'entrée standard jusqu'à ce que l'utilisateur entre 0. À ce moment elle affiche la somme de tous les entiers entrés. Testez-la. Quel est son type ?

11. (*5 points*) Écrivez une fonction qui prend en argument un tableau et sa taille, et le trie.

Fin.