Travaux Dirigés n°2

Cours d'analyse, algorithmique

—Master 2 CCI—

Sous-programmes, fonctions

Cette séance de travaux dirigées est consacrée à l'écriture et à l'utilisation de fonctions. On prendra bien soin de commenter chaque fonction pour préciser son mode d'emploi.

▸ Exercice 1. Fonctions de base

1. Écrire une fonction `factorielle` qui calcule la factorielle d'un entier positif. Pour éviter les problèmes de dépassement de capacité les calculs intermédiaires et le résultat seront des `double`.
2. Écrire une fonction `puissance` qui calcule les puissances entière d'un nombre réel.

▸ Exercice 2. Comparaison de nombres

Sur les nombres à virgule (`float, double`), l'opérateur d'égalité `==` en C/C++ n'est pas très utile à cause des erreurs d'arrondis. De plus, comme le programme n'affiche pas tous les chiffres, deux nombres qui s'affichent de la même manière peuvent être différents. Par exemple, le programme suivant

```
#include<stdio.h>
void main (void) {
  printf("%f, %f \n", (10.0/3.0-3.0)*3.0, 1.0);
  if ((10.0/3.0-3.0)*3.0 == 1.0) printf("Egal\n"); else printf("Pas Egal\n");
}
```

affiche sur ma machine (`gcc (SUSE Linux) 4.5.0 20100604 [gcc-4_5-branch revision 160292]`)

```
1.000000, 1.000000
Pas Egal
```

La différence entre les deux nombres est sur une décimale qui n'est pas affichée. Pour résoudre ce problème, quand on veut comparer deux nombres à virgule, on teste si la valeur absolue de la différence est négligeable devant les deux nombres :

$$|x - y| < \epsilon|x| \qquad \text{et} \qquad |x - y| < \epsilon|y| \tag{1}$$

où ϵ est un très petit nombre.

3. Définir une constante `EPSILON` égale à 10^{-12} ;
4. Écrire une fonction `egal` qui prends deux nombres x et y et qui réponds s'ils vérifient la condition ci-dessus, c'est à dire s'il sont égaux avec une précision de ϵ.

▸ Exercice 3. Saisie contrôlée

5. Écrire une fonction `lisPositif` qui prends en paramètre un caractère représentant le nom d'une variable (par exemple `'x'`), qui affiche le message
   `Donnez la valeur de x :`
   et qui vérifie que le nombre entré est bien un nombre positif. Dans le cas contraire on redemande un nouveau nombre à l'utilisateur.

▸ **Exercice 4. Racine carrée**

On montre en mathématique que étant donné un réel positif a la suite

$$u_0 := a, \qquad u_{n+1} := \frac{u_n + a/u_n}{2} \tag{2}$$

converge vers $\sqrt{a}$.

6. Écrire une fonction qui calcule la racine carrée d'un réel a. Par définition, la racine carrée est la solution x de l'équation $x^2 = a$. On utilisera ce test et la fonction `egal` définie plus haut pour vérifier que l'on a bien le résultat.

▸ **Exercice 5. Exponentielle**

La fonction exponentielle est définie par

$$\exp(a) := \sum_{i=0}^{\infty} a^i/i!\,. \tag{3}$$

7. En réutilisant la fonction `égal` écrire une fonction `exponentielle` qui calcule l'exponentielle d'un nombre a. On utilisera une boucle et un accumulateur pour calculer les sommes $\sum_{i=0}^{N} a^i/i!$. On stoppe la boucle dès que deux sommes calculées consécutivement sont «égales».

Cette méthode n'est pas très efficace car, en utilisant les fonctions `factorielle` et `puissance`, on recalcule plusieurs fois les même produits. Pour aller plus vite, on peut, dans la même boucle, accumuler la factorielle, la puissance et la somme.

8. Écrire une fonction `exponentielle2` qui fait le calcul plus rapidement en utilisant les trois accumulateurs dans la même boucle. On gardera la même condition d'arrêt de la boucle.

▸ **Exercice 6. Logarithme**

Pour calculer le logarithme d'un nombre positif a, on peut utiliser de la même manière que pour la racine le fait que la suite

$$u_0 := a, \qquad u_{n+1} := u_n + \frac{a}{\exp(u_n)} - 1 \tag{4}$$

converge vers $\ln(a)$.

9. Écrire une fonction `logarithme` qui calcule le logarithme d'un nombre positif. Par définition, le logarithme de a est la solution x de l'équation $\exp(x) = a$. On utilisera ce test et la fonction `egal` définie plus haut pour vérifier que l'on a bien le résultat.
10. Écrire un programme qui vérifie que pour un nombre x, on a bien $\exp(\ln(x)) = x$.
11. Écrire un programme qui calcul $\sqrt{x}$ par la formule

$$\sqrt{x} = x^{\frac{1}{2}} = \exp\left(\frac{\ln(x)}{2}\right)\,. \tag{5}$$