Chapitre 3

Les Fonctions

EX. 53 Écrire une fonction à valeur entière qui calcule x exposant i.

EX. 54 Écrire une fonction à valeur booléenne qui teste la parité d'un nombre.

EX. 55 La distance euclidienne entre 2 points de coordonnées (x_1, y_1) et (x_2, y_2) est donnée par la formule :

$$s = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Écrire une fonction qui reçoit les coordonnées de 2 points et calcule la distance entre ces 2 points (vous pouvez utiliser la fonction `sqrt`).

EX. 56 Écrire une fonction à valeur booléenne qui teste la primalité d'un nombre.

EX. 57 Écrire une fonction à valeur entière qui retourne le nombre de nombres premiers strictement inférieurs à une valeur fournie.

EX. 58 Écrire la fonction `swap` qui effectue l'échange de ses deux paramètres entires `a` et `b`.

EX. 59 Écrire la fonction `Fibonacci` qui reçoit en paramètre 2 nombres de Fibonacci consécutifs, disons F_n et F_{n+1} pour fixer les idées. Après l'appel, on souhaite disposer des 2 nombres Fibonacci suivants, c'est-à-dire F_{n+2} et F_{n+3}.

EX. 60 Écrire une fonction qui trie trois variables de manière croissante. Par exemple, considérons les trois variables sont `a`, `b` et `c` contenant respectivement 5, 3 et 7. Après l'appel à la fonction (`trie(a,b,c)`, par exemple), nous aurons $\texttt{a} = 3$, $\texttt{b} = 5$ et $\texttt{c} = 7$.

EX. 61 Chercher les erreurs dans les algorithmes suivants :

(a)
```
// Cette fonction retourne la factorielle de n
int factorielle(int n);
{
  int i:

  for(i=1;i<=N;++i)
  {
    n*=i;
  }
  return(n);
}
```

(b)
```
// Cette fonction multiplie par 2 la valeur contenue dans a.
void fois_2(int a)
{
  a*=2;
}
```

EX. 62 Un algorithme pour trouver le plus grand commun diviseur de deux nombres entiers a été découvert par Silver et Terzian en 1962. Dans beaucoup de cas, il est plus rapide que l'algorithme d'Euclide.

Étant donné deux nombres entiers a et b, l'algorithme procède en trois étapes :

1. Déterminer la plus grande puissance k de 2 qui divise à la fois a et b (où k est un nombre naturel) ; remplacer a par $a/2^k$ et b par $b/2^k$.

2. A présent, a ou b est impair. Si $a \neq b$, faire ce qui suit :
   - $t \leftarrow \mid a - b \mid$
   - Si t est pair, remplacer t par $t/2$. Répéter ceci tant que t est pair.
   - $a \leftarrow t$ si $a > b$, $b \leftarrow t$ sinon.
   - Si $a \neq b$, répéter l'étape 2.

3. à présent, $a = b$. Le plus grand commun diviseur des deux nombres donnés vaut $2^k \cdot a$.

En appliquant cet algorithme à 504 et 420, nous obtenons

a	b
504	420
252	210
126	105
21	105
21	21

Réponse : $2^2 \times 21 = 84$

Écrire une fonction `Silver_Terzian` qui réalise cet algorithme.

: 25 min.

EX. 63 Écrire une fonction qui calcule la valeur approchée de π sur base de la série suivante. Les calculs s'arrêtent lorsque la valeur du dernier terme ajouté est inférieure à un ε donné, ou lorsque le nombre de termes considérés atteint une borne également donnée.

$$\frac{\pi}{4} = 4 \cdot arctan\left(\frac{1}{5}\right) - arctan\left(\frac{1}{239}\right)$$

avec la série :

$$arctan(x) = \frac{x}{1+x^2}\left(1 + \frac{2}{3} \cdot \frac{x^2}{1+x^2} + \frac{2}{3} \cdot \frac{4}{5} \cdot (\frac{x^2}{1+x^2})^2 + \cdots\right) = \frac{x}{1+x^2}\sum_{i=0}^{\infty} c_i(\frac{x^2}{1+x^2})^i$$

où $c_0 = 1$
et pour $i > 0, c_i = \frac{2 \cdot i}{2 \cdot i+1} c_{i-1}$

EX. 64 Deux suites (x_i) et (y_i) sont définies par

$$\begin{aligned} y_0 &= 2 \\ x_0 &= 1 \\ x_{i+1} &= \frac{2}{y_{i+1}} \\ y_{i+1} &= \frac{y_i + x_i}{2} \end{aligned}$$

Nous admettons que les suites convergent toutes les deux vers $\sqrt{2}$, avec de plus $x_i < \sqrt{2} < y_i$. Ecrire une fonction `rac2` qui calcule deux approximations supérieure et inférieure de $\sqrt{2}$, de différence inférieure à ε (prévoyez aussi d'arrêter les calculs si le nombre d'étapes dépasse une borne fixée).

EX. 65 Soit r un nombre réel positif. Deux suites de nombres réels x_n et y_n sont spécifiées par :

$x_1 = 1$,
$y_1 = 1$,

et pour $n \geq 1$:

$$\begin{cases} x_{n+1} = x_n + r.y_n \\ y_{n+1} = x_n + y_n \end{cases}$$

Il n'est pas difficile de prouver $y_n \neq 0$ pour $n \geq 1$, et aussi que le quotient $\frac{x_n}{y_n}$ converge vers $\sqrt{r}$.

Écrire une fonction qui calcule la valeur approchée $\frac{x_n}{y_n}$ de $\sqrt{r}$, où n est la plus petite valeur pour laquelle la condition suivante est vraie :

$(x_n)^2 < (r + \varepsilon)(y_n)^2$ **et** $(x_n)^2 > (r - \varepsilon)(y_n)^2$.

(ε étant comme d'habitude une valeur donnée). L'exécution sera interrompue si le nombre d'étapes devient trop grand.

EX. 66 On souhaite disposer d'un module tools contenant les fonctions `maximum`, `minimum`, `abs` pour des nombres entiers. On vous demande d'écrire ce module, le fichier d'en-tête correspondant et un exemple de module principal qui l'utilise.

EX. 67 Dans certaines conditions, on peut calculer une abscisse où une fonction f s'annule par la méthode de Newton.

Partant d'une approximation initiale x_0 (valeur quelconque qu'il vaut mieux prendre près de l'abscisse cherchée) on calcule les approximations suivantes $x_1, x_2, ..., x_i, x_{i+1}$ par la formule suivante :

$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$

On s'arrête lorsque l'approximation x_i est suffisamment précise.

De cette façon on peut calculer la racine n^{me} d'un nombre a. Il faut calculer la valeur de x pour laquelle la fonction $f(x) = x^n - a$ s'annule.

Ayant $f'(x) = n \cdot x^{n-1}$ on obtient l'évaluation suivante de x :

$x_{i+1} = x_i - \frac{x_i^n - a}{n \cdot x_i^{n-1}}$

On calcule ainsi les approximations successives pour $\sqrt[n]{a}$ jusqu'à ce que :

$|x_i{}^n - a| \leq \varepsilon$ ($\varepsilon = 10^{-5}$ par exemple).

Écrire une fonction racine à valeur réelle qui calcule la racine n^{me} d'un nombre a (a réel et n entier donnés en paramètre). Cette fonction prend $x_0 = 1$.

: 40 min.

EX. 68 Écrire une fonction `pyramide` qui crée une pyramide de hauteur n (n passé comme paramètre) avec les chiffres suivants (à l'aide de boucles imbriquées) :

```
          1
         232
        34543
       4567654
      567898765
     67890109876
    7890123210987
   890123454321098
  90123456765432109
 0123456789876543210
123456789010987654321
```

(Ne pas écrire 11 chaînes de caractères à plusieurs chiffres).
