

Programmation : TP 11 noté (23 mars 2004)

Objectifs : [C, Ocaml] Contrôle de connaissances.

L'utilisation des cours et des TPs précédents est autorisée.

Comme d'habitude, vous rendrez par mail les fichiers (.ml et .c) produits durant votre TP, à adresser à `pascal.brisset@enac.fr`.

[C] Quadrature

Soit un entier positif $n < 10000$. On considère que n s'écrit toujours avec 4 chiffres donc que, par exemple, 12 s'écrit 0012. On crée un nouveau nombre en élevant au carré le nombre obtenu à partir du deuxième et du troisième chiffre de n . On répète cette opération plusieurs fois et on cherche à déterminer en combien d'itérations on retrouve un nombre que l'on a déjà obtenu.

Par exemple, pour $n = 8578$, on obtient 2 fois le nombre 3249 :

n	2 ^e et 3 ^e	nouveau nombre
8578	57	3249
3249	24	0576
0576	57	3249

1. Écrire un programme qui calcule, pour un nombre n , le nombre d'itérations nécessaire pour retrouver un nombre déjà vu.

Indications :

- Il est fortement conseillé d'écrire une ou plusieurs sous-fonctions.
 - On pourra utiliser un tableau de taille fixe afin de stocker les nombres au fur et à mesure qu'on les crée.
2. On souhaite examiner le problème de façon exhaustive. On considère alors toutes les suites u_n définies par :

$$u_{n+1} = 2^{\text{e}} \& 3^{\text{e}} \text{ chiffres de } (u_n)^2$$

avec $u_0 \in [0, 100[$.

Écrire un programme qui, parmi les 100 suites ainsi définies, calcule le nombre de suites convergeant vers 0.

[OCaml] Représentation d'un système de fichiers

Le but est de représenter un système de fichiers et d'implémenter des fonctions classiques de parcours correspondantes.

1. Définir un type permettant de représenter un système de fichiers. On considérera qu'un système de fichiers peut contenir soit des fichiers, caractérisés par leur nom et leur taille respective, soit des répertoires, caractérisés par leur nom et leur contenu respectif.
2. Représenter avec ce type, l'arborescence de la figure 1.
3. Écrire une fonction qui permet de lister de façon récursive le contenu d'un système de fichiers (l'équivalent de la commande `ls -R`).

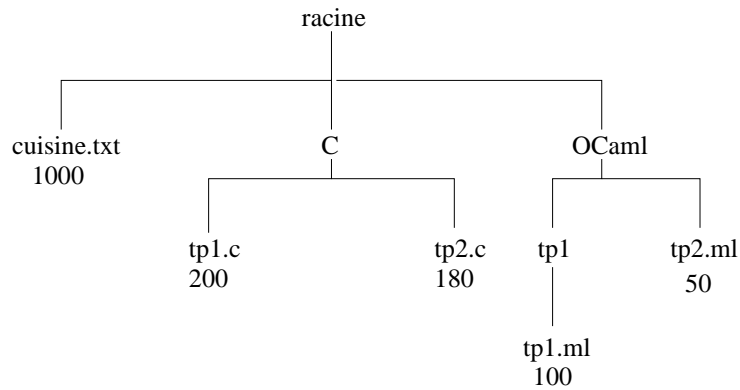


FIG. 1 – Un système de fichiers

Par exemple, l'application de cette fonction sur l'arborescence de la figure 1 affichera quelque chose du genre :

```

racine/
  cuisine.txt
  C/
    tp1.c
    tp2.c
  OCaml/
    tp1/
      tp1.ml
    tp2.ml
  
```

4. Écrire une fonction qui permet d'afficher de façon récursive la taille de chaque répertoire (et sous-répertoire) d'un système de fichiers (l'équivalent de la commande `du`).

Par exemple, l'application de cette fonction sur l'arborescence de la figure 1 affichera quelque chose du genre :

```

380  ./racine/C
100  ./racine/OCaml/tp1
150  ./racine/OCaml
1530 ./racine
  
```