

# Exercices supplémentaires

L1 CUPGE 2013 - Université d'Aix-Marseille

## Exercice 1 (Suite de Syracuse)

Écrivez une fonction `syracuse` prenant en argument un entier  $n$  et effectuant les opérations suivantes :

- Si  $n$  est pair, le diviser par deux ; sinon, le multiplier par trois et ajouter 1 ;
- Afficher la nouvelle valeur de  $n$  ;
- Si  $n \neq 1$ , recommencer.

Écrivez une fonction `main` demandant la valeur de  $n$  à l'utilisateur et appelant la fonction `syracuse`.

## Exercice 2 (Tri bulle)

Écrivez la fonction `void bulle(int t[], int n)` qui parcourt le tableau  $t$  de taille  $n$  et échange la valeur des cases  $t[i]$  et  $t[i+1]$  quand  $t[i] > t[i+1]$ .

Considérons  $t = [5, 1, 4, 3]$ . Quelle est la valeur de  $t$  après 1 passage ? 2 passages ? À votre avis, que va-t'il finir par se passer ?

## Exercice 3 (Décomposition en facteurs premiers)

Écrivez une fonction `void decompose(int k, int facteurs[])` qui remplit le tableau `facteurs` avec les facteurs premiers de  $k$ , dans l'ordre croissant.

## Exercice 4 (Lecture de code)

On suppose qu'on dispose d'une fonction `void aff_tab(int t[], int n)` qui affiche le contenu du tableau  $t$  de taille  $n$  sur une ligne. Considérons le code suivant :

```
void mystere (int tab[], int res[], int n)
{
    int i=1,j=0;
    res[0] = t[0];
    for (i=1 ; i<n ; i++)
    {
        if (t[i] >= res[j])
```

```

    {
        j = j+1;
        res[j] = t[i];
    }
    aff_tab(res,j)
}

```

Qu'est-ce qui est affiché quand `tab` vaut, respectivement :

1; 3; 2; 4; 2; 5 ?

4; 3; 2; 1; 2; 6 ?

À votre avis, que fait cette fonction ?

### Exercice 5 (Introduction à la bioinformatique)

Écrivez un programme `mystere` (`char tab[], int n, char mot[], int m`) qui prend en entrée une grande chaîne de caractères `tab` de longueur  $n$ , une petite `mot` de longueur  $m$ , et qui teste si `mot` apparaît dans `tab`.

### Exercice 6 (Échiquier)

Écrivez une fonction `main` qui définit une matrice d'entiers de taille  $N \times M$  (constantes) qu'on va remplir de 0 et de 1 à la manière d'un échiquier (les 0 et les 1 sont alternés verticalement et horizontalement). Enfin, on affichera la matrice obtenue.

## Entrée-sortie sur des fichiers

### Exercice 7 (Personne n'entend l'espace crier)

On dispose d'un fichier `livre.txt`. Comptez le nombre de mots contenus dans le fichier. Deux mots sont séparés par un espace.

### Exercice 8 (Formation au métier de caissier)

Supposons qu'on dispose d'un fichier `clients.txt`. Sur chaque ligne, on a un numéro de client, suivi d'un espace, suivi de l'argent qu'il doit (en euros, avec deux chiffres après la virgule).

Écrivez une fonction `main` qui demande un numéro de client et affiche son ardoise.

## Plus difficile

### Exercice 9 (Mots mêlés)

Les mots mêlés consistent à trouver des mots dans un carré de lettres, soit horizontalement, soit verticalement, soit en diagonale.

Écrivez une fonction `void demeler (char mat[50][50], char mot[], int n)`, où `mot` est de longueur `n`, et le but est de trouver `mot` à l'intérieur de la matrice `mat`. On affichera la réponse sur le terminal sous la forme :

- Le mot a été trouvé diagonalement, en partant de la case (4,3).
- Le mot n'a pas été trouvé.

### Exercice 10 (No one can be told what the Matrix is. You have to see it for yourself.)

On dispose d'un fichier `matrice.txt` dans lequel est écrit une matrice de la façon suivante :

```
3 9 43 1
8 5 7 16374
4 9 2 0
```

On veut écrire une fonction `main` qui définit une matrice `mat` de la bonne taille (en comptant d'abors le nombre de lignes et de colonnes dans le fichier) et range la matrice à l'intérieur.

Indice : essayez de compter les lignes et les colonnes indépendamment.