

# Introduction à l'informatique

## Exercices — Série 7

### Exercice 1

Écrire une fonction C capable de comparer deux chaînes de caractères données, et de retourner la position du premier caractère qui diffère entre ces deux chaînes (1 s'il s'agit du premier, 2 pour le deuxième, et ainsi de suite), ou bien 0 si les deux chaînes possèdent le même contenu.

### Exercice 2

Écrire une fonction C capable de déterminer si une chaîne de caractères donnée est un palindrome, ce qui signifie que sa lecture de la gauche vers la droite et de la droite vers la gauche fournit le même résultat.

### Exercice 3

Écrire une fonction C capable de calculer la longueur du plus grand suffixe commun entre deux chaînes de caractères données.

### Exercice 4 (*Examen de janvier 2019*)

1. Décrire, le plus simplement possible, l'opération effectuée par la fonction C suivante.

```
unsigned f(char *s)
{
    unsigned n;

    if (!s[0])
        return 0;

    if (s[0] >= '0' && s[0] <= '9')
        n = 1;
    else
        n = 0;

    return n + f(++s);
}
```

2. Quelle est la complexité en temps de cette fonction ?
3. Écrire une fonction C réalisant exactement la même opération, mais sans effectuer d'appel récursif.

**Exercice 5** (*Examen d'août 2019*)

1. Décrire, le plus simplement possible, l'opération effectuée par la fonction C suivante.

```
long f(int t[], int nb)
{
    if (nb <= 0)
        return 0;

    return t[0] + f(t + 2, nb - 2);
}
```

2. Quelle est la complexité en temps de cette fonction ?
3. Écrire une fonction C réalisant exactement la même opération, mais sans effectuer d'appel récursif.