

Contrôle continu LI 012 (Algorithmique)**Devoir sur table n° 1****Aucun document autorisé****Durée : 1 heure 30.**

1. On suppose que l'on dispose d'un texte (enregistré en mémoire sous la forme d'une chaîne de caractères de longueur n) comprenant des « balises », c'est-à-dire des portions de textes encadrées par des séquences de la forme `<xx>` et `</xx>`, où `xx` désigne une suite quelconque de lettres.

Écrire un programme qui, après avoir demandé à l'utilisateur quelle balise l'intéresse (par exemple `table`), extrait (c'est-à-dire affiche à l'écran) toutes les portions de texte encadrée par cette balise (ici, toutes les portions entre `<table>` et `</table>`). On suppose qu'il n'y a pas d'imbrication des balises et que le texte est correctement balisé.

2. On suppose que l'on dispose d'un tableau de chaînes de caractères (de longueur maximale 20 caractères : `string[20]`), de dimension N , correspondant à un ensemble de mots.

```
var Tm = array[1..N] of string[20]
```

Ce tableau n'est pas trié. On veut produire un tableau trié sur l'envers des mots (on considère chaque mot « à l'envers », de la dernière lettre à la première)².

- (a) Écrire une fonction `avant` qui prend en argument deux chaînes de caractère, et répond `vrai` si la première est avant la seconde, dans le sens défini précédemment (c'est-à-dire en prenant les mots à l'envers).

```
function avant(x, y : string[20]) : boolean;
```

- (b) Proposer un algorithme qui réalise le tri du tableau initial, en utilisant la fonction `avant`.
- (c) Discuter et justifier l'algorithme que vous avez choisi.

3. Écrire une fonction récursive qui recherche une valeur dans un tableau d'entiers par dichotomie. On suppose le tableau déjà rempli, de dimension N , et déclaré comme variable globale `tab`.

```
function rech_dicho(val : integer, g,d : integer) : boolean;
```

²On utilise fréquemment ce genre de technique pour travailler sur les suffixes.
