

Contrôle continu & examen final LI 012
Aucun document autorisé.
Durée : 3 heures.

1. Complexité

Soit la fonction C suivante (t est un tableau de caractères, max sa longueur) :

```
for (i=max ; i>0 ; i--) {
  if (t[i] == 'e' || t[i] == 'a') {
    for (j=i ; j<max ; j++) t[j] = t[j+1] ;
    max-- ;
  }
}
```

- (a) Indiquez le déroulement pas à pas du programme avec le tableau t suivant :

m	a	i	s	o	n	\0
---	---	---	---	---	---	----

. Que fait ce programme?
- (b) Calculez la complexité moyenne (en $O()$ par rapport à la valeur de max) de cet algorithme.
- (c) Proposez une façon d'obtenir le même résultat de façon moins coûteuse.

2. Représentation d'ensembles d'entiers sous forme de tableaux de booléens

On définit le type `ensemble` :

```
type ensemble = array[0..MAX] of boolean ;
```

Les variables de type `ensemble` représentent des ensembles d'entiers compris entre 0 et la constante `MAX`.

- (a) **Appartenance.** Écrire la fonction `appartient(ens:ensemble; elt:integer): boolean` qui renvoie `true` si l'entier `elt` appartient à l'ensemble, et `false` sinon.
- (b) **Initialisation.** Écrire la procédure `initialise(var ens : ensemble)` qui initialise l'ensemble `ens` à l'ensemble vide.
- (c) **Cardinal.** Écrire la fonction `cardinal` qui donne le cardinal (nombre d'éléments) de l'ensemble donné en paramètre.
- (d) **Union.** Écrire la procédure `union(e1, e2: ensemble; var e3: ensemble)` qui affecte à `e3` le résultat de l'union de `e1` et `e2`.
- (e) **Intersection.** Écrire la procédure `intersection(e1, e2: ensemble; var e3: ensemble)` qui affecte à `e3` le résultat de l'intersection de `e1` et `e2`.
3. Proposez une implémentation en tableau circulaire d'une **file** de caractères. On suppose que l'on a besoin des primitives *enfiler*, *tête*, *défiler*, *file_vide* et *file_pleine*. On demande une description algorithmique de chaque primitive, sans nécessairement une prise en compte complète de tous les cas particuliers.
-