

# Algorithmique S2 (TI104)

## DE L1 P2020 (2015-16)

17 mai 2016, 14:00-15:45

### Consignes

1. Lisez attentivement le sujet et respectez les modalités à suivre.
2. Sans document, sans calculatrice, sans appareil d'aucune sorte.
3. Réponses concises, claires, complètes, non ambiguës !
4. Pour chaque exercice, argumentez votre analyse et donnez le principe de votre algorithme en une dizaine de lignes maximum, puis produisez votre algorithme en langage algorithmique. Soignez spécialement la présentation de votre algorithme (entête, indentations, etc).
5. Si vous ne savez pas répondre, il vaut mieux ne rien écrire, et se concentrer sur les premiers exercices.
6. Tout étudiant suspecté d'avoir copié ou de s'être inspiré de son voisin sera sévèrement sanctionné.
7. Vous devez respecter les prototypes donnés ci-dessous et compléter leur entête avant de les définir. Vous êtes autorisés - et nous vous recommandons d'exploiter cette possibilité - à définir tous les algorithmes auxiliaires dont vous pourriez avoir besoin dans vos algorithmes principaux.

### Sujet

Pour l'ensemble des exercices qui suivent, vous disposez des deux algorithmes auxiliaires suivants :

1. **Sous-tableau(a : tableau de taille n) : tableau de taille n - 1** qui retourne le sous-tableau constitué des éléments de a sauf le premier (un élément d'index i dans a,  $i > 0$ , a pour index i - 1 dans le sous-tableau).
2. **Taille-tableau(a : tableau de taille n) : entier** qui retourne la taille n de a.

Concevoir et spécifier en langage algorithmique un algorithme qui :

#### A) Factorielle(n : entier naturel) : entier naturel

**ALGORITHME RÉCURSIF** qui retourne la valeur de n!.

#### B) Enregistrer() : Fiche

qui demande à l'utilisateur de saisir ses nom, prénom, âge et retourne une structure Fiche créée dynamiquement et qui comporte ces trois informations.

**NB** - Vous devez définir la structure Fiche avant de l'utiliser dans votre algorithme.

### C) Initialise2D(a : tableau 2D de taille n x n)

**ALGORITHME RÉCURSIF** qui initialise a de manière à ce que la valeur de chaque cellule soit la distance Manhattan (nombre de pas horizontaux ou verticaux) pour atteindre la cellule courante depuis la cellule d'index (n - 1, n - 1). Exemple avec n = 5

```
8 7 6 5 4
7 6 5 4 3
6 5 4 3 2
5 4 3 2 1
4 3 2 1 0
```

### d) Eval(P : tableau de taille n, v : réel) : réel

qui prend en argument un polynôme  $P[X]$  défini en tant que tableau structuré de Monômes (un couple constitué d'un coefficient réel et d'un degré entier naturel) et qui retourne la valeur de  $P[v]$  (l'évaluation de  $P$  en  $X = v$ ).

**NB** - Vous devez définir la structure Monôme avant de l'utiliser dans votre algorithme.

**NB** - Bonus aux étudiants qui produiront une solution récursive qui fonctionne.

### E) Limite(c, z : Complexe) : Complexe

**ALGORITHME RÉCURSIF**, appelé initialement avec un nombre complexe c quelconque et avec  $z = 0$ , qui retourne la limite vers laquelle tend la suite définie par la récurrence suivante :  $Z_{n+1} = Z_n^2 + c$ ,  $Z_0 = 0$ . On suppose (même si cela est faux), que cette suite converge toujours. On considère que la limite est atteinte quand la différence, en module, entre deux termes successifs de la suite, est inférieure à une constante réelle epsilon donnée.

**NB** - Vous devez définir la structure Complexe avant de l'utiliser dans votre algorithme.