

### Quick - 1h

*Calculatrices, téléphones et documents interdits.  
Traductrices autorisées pour les étudiants non francophones.*

## Instructions

Les algorithmes seront **tous** écrits en pseudo-langage algorithmique. On veillera à décrire calmement ce que l'on fait, et à bien indenter les pseudo-codes. Vous indexerez vos tableaux de 0 à  $N - 1$ . On vous fournit des indications de temps largement sur-évaluées.

**Exercice 1** [5 min] Écrire un programme complet qui demande un entier  $x$  à l'utilisateur et imprime tous les entiers entre  $x$  et  $x + 42$  (compris).

**Exercice 2** [5-10min] Écrire un programme complet (**sans utiliser de tableau**) qui demande des entiers strictement positifs à l'utilisateur, s'arrête lorsque l'utilisateur rentre un entier négatif ou nul, et imprime le maximum des entiers positifs entrés.

**Exercice 3** [10-15 min] On considère la suite récurrente suivante :

$$u_n = \begin{cases} 1 & \text{si } n=0 \\ 3 * u_{n-1} + 42 & \text{si } n>0 \end{cases}$$

Écrire un algorithme (fonction ou action ? justifier !) qui calcule et stocke dans un vecteur de taille  $N > 1$  (tableau passé en paramètre modifiable,  $N$  constante fixée supposée supérieure à 1), l'entier  $u_i$  à la case numéro  $i$  du tableau. Quelle est la complexité de votre algorithme en nombre de multiplications ?

**Exercice 4** [30 min] Soit la fonction mystère suivante : ( $N = 3$ )

```

Fonction mystere(i,t) :Booleen
  | D: t :Vecteur[N] d'entiers
  | Si (i<0 ou i=N-1) alors
  | | Retourner Vrai
  | Sinon
  | | Retourner ((t[i]<t[i+1]) et mystere(i+1,t))
  | Fsi
Fonction
  
```

Attention, dans le deuxième **Retourner** il s'agit bien d'une expression booléenne dont on doit connaître le résultat avant de retourner.

Quels calculs réalise l'appel **mystere**(0,*t*) pour  $t = [10, 22, 1]$  ? (avec les étapes) puis pour  $t = [12, 23, 1515]$  (donner uniquement le résultat) ? Dans le cas général ?

## Syntaxe Algorithmique

Nom	Syntaxe	Exemple	Commentaire
Affectation	$\leftarrow$	$x \leftarrow 42$	$x$ doit être déclaré
Type entier	Entier	$x : \text{Entier}$	déclaration de $x$ entier
Type réel	Réel	$x : \text{Réel}$	déclaration d'un réel, en machine ce sera un flottant
Type caractère	Caractère	$c : \text{Caractère}$	déclaration d'un caractère; les constantes sont 'a', 'b', ...
Type booléen	Booléen	$b : \text{Booléen}$	déclaration d'un booléen; les constantes sont <b>Vrai</b> et <b>Faux</b>
Type chaîne	Chaîne	$s \leftarrow \text{"toto"}$	affectation d'une chaîne, $s$ doit être déclarée.
Tableau	Vecteur	Vecteur[10] d'Entiers	tableau de 10 entiers indexés de 0 à 9
constante	Constante	Constante $N : 10$	déclare une constante symbolique $N$ qui vaut "10"

### Tests

**Si** *condition* **alors**  
 | instructions si vrai  
**Sinon**  
 | instructions si faux  
**Fsi**

**Si** *condition* **alors**  
 | instructions si vrai  
**Fsi**

### Boucles

**Pour**  $i$  **de** *inf* **à** *sup* **Faire**  
 | instructions  
**Fpour**

**Tq** *condition* **faire**  
 | instructions  
**Ftq**

### Programme/fonction/action : exemples

**Programme**  
*Main()*  
 ....  
 ....  
 ....  
**Retourner** 0  
**FProgramme**

**Fonction** *fonct(c) : Entier*  
**D**:  $c : \text{Caractère}$   
**L**:  $s : \text{Entier}$   
 ....  
 ....  
 ....  
**Retourner**  $s$   
**FFonction**

**Action** *monact(a,b,c)*  
**D**:  $a : \text{Entier}$   
**D/R**:  $b : \text{Entier}$   
**R**:  $c : \text{Caractère}$   
 ...  
 ...  
 ...  
**FAction**

- ici *fonct* est une fonction **Caractère**  $\rightarrow$  **Entier** :  
 L'unique paramètre (la *donnée*) est un caractère, nommé  $c$  dans la suite de la fonction. La variable  $s$  est *locale*. L'appel :  $resu \leftarrow \text{fonct}(d)$  où  $d$  a une valeur et  $resu$  est déclaré de bon type.
- ici *monact* est une action à trois arguments. Les modifications apportées au premier paramètre ne sont pas enregistrées (c'est une *donnée*). Par contre les modifications apportées aux paramètres 2 et 3 sont enregistrées (*donnée-résultat* et *résultat*). Appel :  $\text{monact}(a, b, c)$  avec  $a, b$  ayant une valeur, et  $c$  étant déclaré.