

Option Algorithmique et programmation

Corrigé de l'examen de Mai 2007

1. Décision On suppose déclarées et initialisées les variables n, a, b, c de type *integer*, f, g, h de type *boolean*.

1.1 Expliquer ce qu'affiche l'instruction suivante en fonction des valeurs initiales de f, g, h .

```
IF f AND (g OR h) THEN PUT ("un");
ELSE IF g THEN PUT ("deux");END IF; IF h THEN PUT ("trois");END IF;
END IF;
```

Ici on fait bien attention après ELSE il y a deux instructions de type IF THEN donc il peut y avoir deux actions exécutées dans ce cas. De plus ces instructions n'ayant pas de ELSE il est aussi possible qu'aucune action ne soit exécutée.

On peut faire un tableau suivant les valeurs initiales de f, g, h

f	g	h	affichage
0	0	0	
0	0	1	trois
0	1	0	deux
0	1	1	deux trois
1	0	0	
1	0	1	un
1	1	0	un
1	1	1	un

1.2 Ecrire une suite d'instructions qui affecte à n la plus grande des trois valeurs a, b, c .

```
IF a > b THEN IF a > c THEN n := a; ELSE n :=c;END IF;
ELSE IF b > c THEN n := b; ELSE n :=c;END IF;END IF;
```

2. Itération On suppose déclarées n, k, p de type *integer*.

2.1 Expliquer ce que valent les variables n, k, p après la suite d'instructions suivante

```
p := 1; n := 2; k := 9;
LOOP EXIT WHEN k=0;
IF k REM 2=1 THEN p := p * n;END IF;
n := n * n; k := k / 2;
END LOOP;
```

On peut faire un tableau de variations des valeurs des variables

$k = 0 ?$	p	n	k
	1	2	9
F	2	4	4
F	2	16	2
F	2	256	1
F	512	65536	0
V(arrêt de l'itération)			

2.2 *Ecrire une suite d'instructions qui calcule la somme s des diviseurs de n . On n'oubliera pas d'initialiser s (par exemple si n vaut 10, s vaudra $10+5+2+1=18$).*

$s := 0$;

FOR i IN 1.. n LOOP

 IF $n \text{ REM } i = 0$ THEN $s := s+i$; END IF; -- si n est divisible par i alors on ajoute i à s

END LOOP;

3. Tableaux Soient les déclarations suivantes

n : **constant integer** := 1000;

TYPE t_tab IS ARRAY (1.. n) OF **integer**;

$g, d, sauv$: **integer**; t : t_tab ;

3.1 *Ecrire une fonction Min qui a pour paramètre un tableau t et qui renvoie la plus petite valeur contenue dans t .*

FUNCTION $Min(t : t_tab)$ RETURN **integer** IS

m : **integer**;

BEGIN

$m := t(t'first)$; -- le minimum est pour l'instant la première valeur

 FOR i IN $t'range$ LOOP -- i parcourt les indices de numérotation du tableau

 IF $t(i) < m$ THEN $m := t(i)$; END IF;

-- si la valeur d'indice i est inférieure au minimum de celles que l'on a lues

-- (jusqu'à l'indice $i-1$) alors c'est elle qui devient le minimum jusqu'à l'indice i

 END LOOP;

 RETURN m ; -- la valeur renvoyée est la valeur finale de m

END Min ;

3.2 *Expliquer ce que fait la suite d'instructions suivantes*

```
g := t'FIRST ; d := t'LAST ;
```

```
LOOP
```

```
  LOOP
```

```
    EXIT WHEN t(g) >= 0 OR g = t'last ;
```

```
    g := g+1 ;
```

```
  END LOOP ;
```

- - *g parcourt t depuis le premier indice et*

- - *s'arrête sur une valeur positive si elle existe ou bien sur le dernier indice*

```
  LOOP
```

```
    EXIT WHEN t(d) < 0 OR d = t'first ;
```

```
    d := d-1 ;
```

```
  END LOOP ;
```

- - *d parcourt t depuis le dernier indice vers le premier et*

- - *s'arrête sur une valeur négative si elle existe ou bien sur le premier indice*

```
    EXIT WHEN g >= d ;
```

- - *si g et d se sont croisés on arrête*

```
    sauv := t(g) ; t(g) := t(d) ; t(d) := sauv ;
```

- - *si g est bien à gauche de d (on n'est pas sorti de l'itération)*

- - *alors on échange les valeurs de t aux indices g et d*

- - *ce qui a pour effet de déplacer une valeur positive qui était à gauche d'une valeur négative*

- - *à droite de cette dernière*

```
    g := g+1 ; d := d-1 ;
```

- - *l'itération repart, g ayant avancé d'un rang et d reculé d'un rang*

```
  END LOOP ;
```

On a donc déplacé toutes les valeurs négatives à gauche du tableau et toutes les valeurs positives à droite du tableau (cf drapeau de Paris)