

# TD - Bases d'algorithmique

## Exercice 1

1. Addition, soustraction, multiplication, division, modulo.
2. Les flottants servent à représenter les nombres réels. Tous les réels ne peuvent pas être représentés en machine, c'est donc une approximation qui est stockée : les opérations avec les flottants ne sont pas exactes.
3. Un booléen représente une valeur de vérité : VRAI ou FAUX.  
les opérations booléennes sont la disjonction OU, conjonction ET, négation NON.
4. Concaténer deux chaînes de caractères consiste à les coller ensemble pour former une unique chaîne. Par exemple, concaténer "ab" et "cd" donne la chaîne "abcd".  
Dupliquer une chaîne consiste à répéter cette chaîne un certain nombre de fois pour former une unique chaîne. Par exemple, dupliquer "ab" trois fois donne la chaîne "ababab".
5. =, ≠, <, ≤, >, ≥
6. Une exception est déclenchée lorsqu'un problème survient pendant l'exécution d'une instruction. Si  $x < 0$ , alors l'expression  $2/0 = 1$  est évaluée et une exception est déclenchée car on ne peut pas diviser par 0. Si non ( $x \geq 0$ ), l'expression  $x < 0$  est évaluée à FAUX, or "FAUX ET n'importe quoi" est toujours FAUX, donc l'expression  $2/0 = 1$  n'est pas évaluée, donc il n'y a pas d'exception.
7.  $a = 2, b = 4$        $x = 2, y = 1, z = 2$        $a = 1, b = 1, c = 4$
8.  $c \leftarrow a$   
 $a \leftarrow b$   
 $b \leftarrow c$

```

9. chaîne1 ← "oh "
coeff ← 3
chaîne 2 ← coeff × chaîne1
chaîne 2 ← chaîne2 + "!"
```

Pour le 2<sup>ème</sup> programme, on place les deux instructions en trop à la fin.

## Exercice 2

1. FONCTION donne\_saison (jour, mois) :

```
Si (mois = 12 ET jour ≥ 21) OU mois = 1 OU mois = 2 OU
(mois = 3 ET jour < 21) ALORS
    saison ← "hiver"
```

```
SINON Si (mois = 3 ET jour ≥ 21) OU mois = 4 OU
mois = 5 OU (mois = 6 ET jour < 21) ALORS
    saison ← "printemps"
```

```
SINON Si (mois = 6 ET jour ≥ 21) OU mois = 7 OU
mois = 8 OU (mois = 9 ET jour < 21) ALORS
    saison ← "été"
```

```
SINON
    saison ← "automne"
```

FIN SI

RENOVYER saison

FIN FONCTION

2. FONCTION bissextile (année) :

```
Si année MODULO 400 = 0 ALORS
    RENVOYER VRAI
```

```
SINON Si année MODULO 4 = 0 ET année MODULO 100 ≠ 0 ALORS
    RENVOYER VRAI
```

SINON

RENOVYER FAUX

FIN SI

FIN FONCTION

3. FONCTION somme ( $n$ ) :

```
res ← 0  
POUR i ALLANT DE 0 À n FAIRE  
    res ← res + i  
FIN POUR  
RENOYER res  
FIN FONCTION
```

4. FONCTION puissance ( $n, p$ ) :

```
res ← 1  
POUR i ALLANT DE 1 À p FAIRE  
    res ← res × n  
FIN POUR  
RENOYER res  
FIN FONCTION
```

5. FONCTION nombre\_voyelles (-chaine) :

```
res ← 0  
POUR car ALLANT DE premier-caractère AU dernier-caractère FAIRE  
    Si car = 'a' OU car = 'e' OU car = 'i' OU car = 'o' OU  
        car = 'u' OU car = 'y' ALORS  
            res ← res + 1  
        FIN SI  
    FIN POUR  
RENOYER res  
FIN FONCTION
```

6. FONCTION palindrome (-chaine) :

```
a_l_envers ← ""  
POUR car ALLANT DE premier-caractère AU dernier-caractère FAIRE  
    a_l_envers ← car + a_l_envers  
FIN POUR  
RENOYER chaine = a_l_envers  
FIN FONCTION
```

7. FONCTION depense\_5000 () :

```
i ← 1  
produit ← 1  
TANT QUE produit ≤ 5000 FAIRE  
    i ← i + 1  
    produit ← produit × i  
RENOYER i  
FIN FONCTION
```

8. FONCTION cheveux () :

```
longueur ← 21  
jours ← 0  
TANT QUE longueur < 50 FAIRE  
    longueur ← longueur × 1,01  
    jours ← jours + 1  
FIN TANT QUE  
RENOYER jours  
FIN FONCTION
```

9. FONCTION premier ( $p$ ) :

```
POUR i ALLANT DE 2 A p - 1 FAIRE  
    Si p MODULO i = 0 ALORS  
        RENVOYER FAUX  
    FIN SI  
FIN POUR  
RENOYER VRAI  
FIN FONCTION
```

Exercice 3

1. FONCTION somme\_tableau ( $T$ ) :

```
res ← 0  
POUR i ALLANT DE 0 À TAILLE(T) - 1 FAIRE  
    res ← res + Ti  
FIN POUR  
RENOYER res  
FIN FONCTION
```

2. FONCTION nombre\_occurrences ( $T$ , valeur) :

```
res ← 0  
POUR i ALLANT DE 0 À TAILLE( $T$ ) - 1 FAIRE  
    Si  $T_i$  = valeur ALORS  
        res ← res + 1  
    FIN SI  
FIN POUR  
RENVOYER res  
FIN FONCTION
```

3. FONCTION premiere\_occurrence ( $T$ , valeur) :

```
POUR i ALLANT DE 0 À TAILLE( $T$ ) - 1 FAIRE  
    Si  $T_i$  = valeur ALORS  
        RENVOYER i  
    FIN SI  
FIN POUR  
FIN FONCTION
```

4. FONCTION dernière\_occurrence ( $T$ , valeur) :

```
res ← -1  
POUR i ALLANT DE 0 À TAILLE( $T$ ) - 1 FAIRE  
    Si  $T_i$  = valeur ALORS  
        res ← i  
    FIN SI  
FIN POUR  
RENVOYER res  
FIN FONCTION
```

5. FONCTION maximum ( $T$ ) :

```
res ← -∞  
POUR i ALLANT DE 0 À TAILLE( $T$ ) - 1 FAIRE  
    Si  $T_i$  > res ALORS  
        res ←  $T_i$   
    FIN SI  
FIN POUR  
RENVOYER res  
FIN FONCTION
```

6. FONCTION indice\_minimum ( $T$ ) :

```
min ← +∞  
indice_min ← -1  
POUR i ALLANT DE 0 À TAILLE( $T$ ) - 1 FAIRE  
    Si  $T_i$  < min ALORS  
        min ←  $T_i$   
        indice_min ← i  
    FIN SI  
FIN POUR  
RENVOYER indice_min  
FIN FONCTION
```