

# Algorithmique et structures de données 2017-18

## Contenu du cours 4 du 12.10.2017

### 1. Plus Petit Commun Multiple (PPCM)

- Algorithme du PPCM : exemple avec 12 et 15

- Test  $>$ ,  $<$ ,  $=$

- 12  $<$  15

- 24  $>$  15

- 24  $<$  30

- 36  $>$  30

- 36  $<$  45

- 48  $>$  45

- 48  $<$  60

- 60 = 60 Fin !

- Fonction PPCM: `function PPCM(N, M : Positive) return Positive;`

### 2. Plus Grand Commun Diviseur (PGCD)

- Algorithme naïf du PGCD

- Pseudo-code du calcul du PGCD de N et M

- pour I de  $\sqrt{N}$  à 1 faire

- si N rem I = 0 et M rem I = 0 alors

- PGCD := I

- exit

- fin si

- fin pour

- Algorithme d'Euclide : exemple avec N = 35 et M = 60

- Dividende = Diviseur \* Quotient + Reste

- 35 = 60 \* 0 + 35

- 60 = 35 \* 1 + 25

- 35 = 25 \* 1 + 10

- 25 = 10 \* 2 + 5

- 10 = 5 \* 2 + 0

- PGCD = 5

- Fonction PGCD: `function PGCD(N, M : Positive) return Positive;`

### 3. Exercice : Montrer que $\text{PGCD}(M,N) * \text{PPCM}(M,N) = M * N$ en utilisant la décomposition en facteurs 1<sup>er</sup>

### 4. Le tri par insertion

- Algorithme appliqué au tableau

4	7	6	1	2
---	---	---	---	---

4	7	6	1	2
---	---	---	---	---

1<sup>ère</sup> étape

- trouver la position I de 7 dans le sous-tableau trié à 1 élément : 

4
---
- décaler de 1 les éléments de la position I=2 à la fin du sous-tableau
- insérer l'élément 7 en position I=2

4	7	6	1	2
---	---	---	---	---

### 2<sup>ème</sup> étape

- trouver la position I de 6 dans le sous-tableau trié à 2 éléments : 

4	7
---	---
- décaler de 1 les éléments de la position I=2 à la fin du sous-tableau
- insérer l'élément 6 en position I=2

4	6	7	1	2
---	---	---	---	---

### 3<sup>ème</sup> étape

- trouver la position I de 1 dans le sous-tableau trié à 3 éléments : 

4	6	7
---	---	---
- décaler de 1 les éléments de la position I=1 à la fin du sous-tableau
- insérer l'élément 1 en position I=1

1	4	6	7	2
---	---	---	---	---

### 4<sup>ème</sup> étape

- trouver la position I de 2 dans le sous-tableau trié à 4 éléments : 

1	4	6	7
---	---	---	---
- décaler de 1 les éléments de la position I=2 à la fin du sous-tableau
- insérer l'élément 2 en position I=2

1	2	4	6	7
---	---	---	---	---

### • Code en Ada

```
type T_Stat is array(Positive range <>) of Float;
function Position(X : T_Stat) return Positive is ...;
procedure Tri_Insertion(X : in out T_Stat) is
  Pos : Positive;
  Tmp : Float;
begin
  for J in X'first+1..X'last loop
    Pos := Position(X(X'first..J-1));
    Tmp := X(J);
    X(Pos+1..J) := X(Pos..J+1); -- décalage
    X(Pos) := Tmp;
  end loop;
end Tri_Insertion;
```

### 5. Type article (record ou enregistrement)

- Exemple de type pour une date

```
subtype T_Jour is Integer range 1..31;
type T_Mois
  is (Jan,Fev,Mar,Avr,Mai,Juin,Juil,Aout,Sept,Oct,Nov,Dec);
type T_Date is record
  Jour : T_Jour;
  Mois : T_Mois;
  An : Integer;
end record;
```

- Syntaxe

```
type T_Date is record
  Champ_1 : Type_1 := Expression_1;
  Champ_2 : Type_2;
  Champ_3 : Type_3;
```

```

...
end record;

```

- Exemple pour une fraction

```
type T_Fraction is record
  Num : Integer;
  Den : Positive;
end record;
```

- Fonction ou procédure de réduction d'une fraction

```
function Reduire(Frac : T_Fraction) return T_Fraction;
procedure Reduire(Frac : in out T_Fraction);
```

- Exemple d'utilisation d'agrégats avec les fractions

```
F1 : T_Fraction := (2,3); -- F1 = 2/3
F2 : T_Fraction := (Num => 2, Den => 3); -- F2 = 2/3
F3 : T_Fraction := (Den => 2, Num => 3); -- F3 = 3/2
F4 : T_Fraction := T_Fraction'(5,6); -- pour lever les ambiguïtés
                                     -- potentielles
```

- Comparaison avec  $=$  et  $\neq$  se fait automatiquement champ par champ

- Accès aux champs (exemple des fractions)

```
F1.Num := 2;  
F1.Den := 3;  -- F1 = 2/3, idem que F1 := (2,3);  
Put(F1.Num);  
F2.Num := F2.Den;  
Get(F3.Den);  
F4 := (F1.Den, F1.Num);
```

- Annuaire

```

type T_Sexe is (Masc, Fem);
type T_Personne is record
  Nom : String(1..12);
  Sexe : T_Sexe;
  No_Tel : Natural:= 0;
end record;

type T_Annuaire is array(Positive range <>) of T_Personne;
Annu : T_Annuaire(1..3000):= (1 => (Nom => "Paul Martin ",
                                     Sexe => Masc,
                                     No_Tel => 3380410),
                              others => (Sexe => Masc,
                                         No_Tel => 3380410,
                                         Nom => "Toto"
                                         ));

```

## 6. Surcharge d'opérateurs

- Exemple des fractions pour l'opérateur "+"

```
function "+"(F1,F2 : T_Fraction) return T_Fraction is
begin
    return Reduire(T_Fraction'(F1.Num*F2.Num,F1.Den*F2.Den));
end "+";
```

- Exemple des dates pour l'opérateur "<"

```
function "<"(D1,D2 : T_Date) return Boolean is
begin
    if D1.An < D2.An then
        return True;
    elsif D1.An = D2.An and D1.Mois < D2.Mois then
        return True;
    elsif D1.An = D2.An and D1.Mois = D2.Mois
        and D1.Jour < D2.Jour then
        return True;
    else
        return False;
    end if;
end "<";
```