

2 Structures conditionnelles

2.1 Exemple simple

- Quel affichage va être produit par cet algorithme ?

```
VARIABLES
  A : nombre réel
  B : nombre réel
DEBUT
  A ← 5
  B ← A
  SI (A > B)
  ALORS
    ECRIRE (A est plus grand que B)
  FINSI
FIN
```

Cet algorithme n'affiche rien, car la condition $A > B$ est fausse

2.2 Conditions multiples

- Quel affichage va être produit par cet algorithme ?

```
VARIABLES
  a : nombre réel
  b : nombre réel
  c : nombre réel
DEBUT
  a ← 5
  b ← 10
  c ← 15

  # Test de type ET
  SI (a > b) ET (a < c) ALORS
    ECRIRE (A est plus grand que B ET plus petit que C)
  FINSI
```

Rien ne s'affiche, car A est plus petit que B

```
  # Test de type OU
  SI (a > b) OU (a < c) ALORS
    ECRIRE (A est plus grand que B OU plus petit que C)
  FINSI
```

L'algorithme affiche « A est plus grand que B OU plus petit que C », car A est plus petit que C

```
  # Test avec négation
  SI NON ((a > b) ET (a < c)) ALORS
    ECRIRE (la condition est vraie)
  SINON
    ECRIRE (la condition est fausse)
  FINSI
```

L'algorithme affiche « la condition est vraie »

```
  # Test multiple
  SI NON ((a > b) ET (a < c)) ET (b < c) ALORS
    ECRIRE (la condition est vraie)
  SINON
    ECRIRE (la condition est fausse)
  FINSI
```

L'algorithme affiche « la condition est vraie »

```
FIN
```

2.3 Tri de chaînes de caractères

- Ecrire un algorithme qui demande 2 chaînes de caractères, puis les affiche par ordre alphabétique

ALGORITHME Tri de chaînes de caractères

VARIABLES

Chaine1 : chaîne de caractères
Chaine2 : chaîne de caractères

DEBUT

LIRE(chaine1)
LIRE(chaine2)
SI (chaine1 < chaine2) **ALORS**
 ECRIRE(chaine1 chaine2)
SINON
 ECRIRE(chaine2 chaine1)
FINSI

FIN

2.4 Produit de 2 nombres

- Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si son produit est négatif ou positif (on laisse de côté le cas où le produit est nul).
- Faire un 2ème algorithme qui ne passe pas par le calcul du produit des deux nombres.
- Faire un 3ème algorithme, qui cette fois-ci prend en charge le cas où le produit est nul.

ALGORITHME ProduitDe2Nombres

VARIABLES

a : nombre réel
b : nombre réel
resultat: chaîne de caractères

DEBUT

LIRE(a)
LIRE(b)
SI (a = 0) **OU** (b = 0) **ALORS**
 Résultat ← 'Nul'
SINON
 SI ((a < 0) **ET** (b > 0)) **OU** ((a > 0) **ET** (b < 0)) **ALORS**
 # Autre possibilité **SI** ((a < 0) **XOR** (b < 0))
 # Ou encore ((a < 0) **OU** (b < 0)) **ET** **NON**((a < 0) **ET** (b < 0))
 resultat ← 'Négatif'
 SINON
 resultat ← 'Positif'
 FINSI
FINSI
ECRIRE(resultat)

FIN

2.5 Est-ce que je suis le(la) meilleur(e) ?

- Ecrire un algorithme qui demande votre note, ainsi que celle de votre voisin, et qui ensuite vous dit si vous êtes le(la) meilleur(e).

ALGORITHME LeOuLaMeilleurE

VARIABLES

manote : nombre réel
notedemonvoisin : nombre réel
resultat : chaîne de caractères

DEBUT

LIRE(manote)
LIRE(notedemonvoisin)
SI (manote = notedemonvoisin) **ALORS**
 resultat ← 'On est tous les deux aussi bons'
SINON
 SI (manote < notedemonvoisin) **ALORS**
 resultat ← 'Bon, je ferai mieux la prochaine fois...'
 SINON

```

        resultat ← 'Je suis le(la) meilleur(e)!!!'
    FINSI
FINSI
    ECRIRE(resultat)
FIN

```

2.6 Catégorie d'âge

- Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :
 - o "Poussin" de 6 à 7 ans
 - o "Pupille" de 8 à 9 ans
 - o "Minime" de 10 à 11 ans
 - o "Cadet" après 12 ans
- Donner 2 algorithmes, l'un avec **SI...ALORS...SINON**, l'autre avec **AU CAS OU...**
- Dans un deuxième temps, ajouter des éléments pour gérer les cas où l'âge d'un enfant ne correspond à aucune de ces catégories.

La première version de l'algorithme **CategorieAge** ressemble à ça :

```

ALGORITHME CategorieAge
VARIABLES
    age : entier
    categorie : chaîne de caractères
DEBUT
    LIRE(age)
    SI (age >= 6) et (age <= 7) ALORS
        categorie ← 'Poussins'
    SINON SI (age >= 8) et (age <= 9) ALORS
        categorie ← 'Pupille'
    SINON SI (age >= 10) et (age <= 11) ALORS
        categorie ← 'Minime'
    SINON SI (age > 11) ALORS
        categorie ← 'Cadet'
    SINON
        categorie ← 'Hors categorie'
    FINSI
    ECRIRE(categorie)
FIN

```

On peut faire cela aussi en version imbriquée mais c'est déconseillé car difficile à lire :

```

ALGORITHME CategorieAge
VARIABLES
    age : entier
    categorie : chaîne de caractères
DEBUT
    LIRE(age)
    SI (age >= 6) et (age <= 7) ALORS
        categorie ← 'Poussins'
    SINON
        SI (age >= 8) et (age <= 9) ALORS
            categorie ← 'Pupille'
        SINON
            SI (age >= 10) et (age <= 11) ALORS
                categorie ← 'Minime'
            SINON
                SI (age > 11) ALORS
                    categorie ← 'Cadet'
                SINON
                    categorie ← 'Hors categorie'
                FINSI
            FINSI
        FINSI
    FINSI
    ECRIRE(categorie)
FIN

```

La version AU CAS OU donne l'algorithme suivant :

```
ALGORITHME CategorieAge
VARIABLES
    age : entier
    categorie : chaîne de caractères
DEBUT
    LIRE(age)
    AU CAS OU age VAUT
        (age >= 6) et (age <= 7) ALORS categorie ← 'Poussins'
        (age >= 8) et (age <= 9) ALORS categorie ← 'Pupille'
        (age >= 10) et (age <= 11) ALORS categorie ← 'Minime'
        (age > 11) ALORS categorie ← 'Cadet'
    SINON
        categorie ← 'Hors categorie'
    FIN DE CAS
    ECRIRE(categorie)
FIN
```

2.7 Calculatrice simple

- Ecrire l'algorithme d'une calculatrice simple, qui permet de saisir 2 nombres, de choisir l'opérateur désiré (addition, soustraction, multiplication, division) et qui affiche le résultat du calcul.

```
ALGORITHME CategorieAge
VARIABLES
    num1 : nombre réel
    num2 : nombre réel
    operation : nombre entier #ou chaine de caractères, dépend de l'algo choisi
    resultat : nombre réel
DEBUT
    LIRE(num1)
    LIRE(num2)
    LIRE(operation)

    SI (operation = "+") ALORS
        resultat ← num1 + num2
    SINON SI (operation = "-") ALORS
        resultat ← num1 - num2
    SINON SI (operation = "*") ALORS
        resultat ← num1 * num2
    SINON SI (operation = "/") ALORS
        SI (num2 = 0) ALORS
            resultat ← 'Division impossible'
        SINON
            resultat ← num1 / num2
        FINSI
    SINON
        resultat ← 'ERREUR'
    FINSI
    ECRIRE(resultat)
FIN
```

Comme dans le cas précédent l'algorithme peut être réalisé également à l'aide de AU CAS OU

```
ALGORITHME CategorieAge
VARIABLES
    num1 : nombre réel
    num2 : nombre réel
    operation : nombre entier #ou chaine de caractère, dépend de l'algo choisi
    resultat : nombre réel
DEBUT
    LIRE(num1)
    LIRE(num2)
    LIRE(operation)
    AU CAS OU operation VAUT
        "+" ALORS resultat ← num1 + num2
        "-" ALORS resultat ← num1 - num2
```

```

    "*" ALORS resultat ← num1 * num2
    "/" ALORS
        SI (num2 = 0) ALORS
            resultat ← 'Division impossible'
        SINON
            resultat ← num1 / num2
        FINSI
    SINON
        resultat ← 'ERREUR'
    FINSI
    ECRIRE(resultat)
FIN

```

2.8 Photocopies

- Un magasin de reprographie facture 0,10 E les dix premières photocopies, 0,09 E les vingt suivantes et 0,08 E au-delà. Ecrivez un algorithme qui demande à l'utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

```

ALGORITHME Photocopies
VARIABLES
    nbcopies : nombre entier
    apayer : nombre réel
DEBUT
    LIRE(nbcopies)
    SI (nbcopies <= 10)
        ALORS
            apayer ← nbcopies * 0,10
        SINON
            SI (nbcopies <= 30)
                ALORS
                    # 0.10 € pour les 10 premières, puis 0.09 € pour les suivantes
                    apayer ← 10 * 0,10 + (nbcopies - 10) * 0,09
                SINON
                    # 0.10 € pour les 10 premières, puis 0.09 € pour les 20 suivantes, puis 0.08 pour les suivantes
                    apayer ← 10 * 0,10 + 20 * 0.09 + (nbcopies - 30) * 0,08
                FINSI
            FINSI
        ECRIRE(Vous aurez à payer la somme de apayer € pour ces nbcopies photocopies)
FIN

```

2.9 Monnaie

- Ecrire un algorithme qui à partir d'une somme d'argent donnée, la décompose en un nombre minimal de billets et de pièces (rappel des valeurs disponibles en € : 500, 200, 100, 50, 20, 10, 5, 2, 1, 0.50, 0.20, 0.10, 0.05, 0.02 et 0.01). La somme initiale sera saisie par l'utilisateur

Voici un exemple de cet algorithme en python :

```

x=[500,200,100,50,20,10,5,2,1,0.5,0.2,0.1,0.05,0.02,0.01]
y = list(range(15))          # y = [0,1,3,4,5,6,7,8,9,10,11,12,13,14]
n = 0
somme = float(input("Veuillez donner une somme d'argent pour voir la décomposition :"))
num = round(somme,2)         # num = float("{0:.2f}".format(somme))

for i in range(len(x)):      # len(x) donne la taille de la liste
    y[i] = int(num//x[i])     # // donne un entier comme le résultat de division
    num = round(num%x[i],2)   # % (modulo) donne le reste de division par //

print("Pour la somme de ",somme," euros cela donne :")

for i in range(len(y)):
    if(y[i] != 0):
        print(x[i]," ",y[i]," ; ")

```