LIFAP1 – TD 1 : Algorithmes simples

Objectifs : Manipuler les notions de bases vues en CM 1
Définition de type, variable
Instruction, séquence d'instructions
Gestion des entrées / sorties
Structures de contrôle : condition, boucle, …

Les instructions seront écrites uniquement **en langage algorithmique**

Séquence / bloc Début … Fin **Affectation** a ← 5	**Entrées /sorties** utilisateur ➔ machine Lire (valeur) / Saisir (Valeur) utilisateur ← machine Afficher (valeur) / Ecrire (Valeur)
Conditionnelle Si condition alors instruction(s) Sinon instruction(s) FinSi	**Boucle inconditionnelle** Pour i allant de … à … par pas de … faire instruction(s) FinPour
Boucle conditionnelle	
TantQue condition faire instruction(s) FinTantQue	Faire instruction(s) TantQue condition

1. Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

```
Début
A, B, C : Entier
   A ← 3
   B ← 10
   C ← A + B
   B ← A * C
   A ← C + 4
Fin
```

Après	La valeur des variables est :		
A ← 3	A = 3	B = ?	C = ?
B ← 10	A = 3	B = 10	C = ?
C ← A + B	A = 3	B = 10	C = 13
B ← A * C	A = 3	B = 39	C = 13
A ← C + 4	**A = 17**	**B = 39**	**C = 13**

2. Écrire un algorithme qui demande un nombre entier à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.
   Exemple : valeur saisie : 5 ➔ résultat affiché : 25

```
Début
  valeur, carre : entier
  Afficher('Donnez la valeur dont vous voulez calculer le carré')
  Saisir(valeur)
  carre ← valeur*valeur
  Afficher('le carré de', valeur, 'est', carre)
Fin
```

Bien constater la différence dans les affichages : chaîne de caractères / contenu de variable.

3. Écrire un algorithme qui demande deux nombres entiers à l'utilisateur et l'informe ensuite si le produit est négatif, positif ou nul. Attention, on ne doit pas calculer le produit !

```
Début
    N1, N2 : Entiers
    Afficher ('Entrez nombre 1 :')
    Saisir (N1)
    Afficher ('Entrez nombre 2 :')
    Saisir (N2)
    Si ((N1 < 0 et N2 < 0) ou (N1 > 0 et N2 > 0))
            Alors   Afficher ('Le produit est positif.')
            Sinon   Si ((N1 < 0 et N2 > 0) ou (N1 > 0 et N2 < 0))
                        Alors Afficher ('Le produit est négatif.')
                        Sinon Afficher ('Le produit est nul.')
                    Fin si
    Fin si
Fin
```

4. Écrire l'algorithme d'un programme permettant d'afficher la table de multiplication d'un entier saisi par l'utilisateur.
   Exemple : valeur saisie : 5 ➔ résultat affiché : 0 5 10 15 20 25 30 35 40 45 50

```
Début
  N, i : Entier
  Afficher('Entrez un nombre')
  Saisir(N)
  Afficher('La table de multiplication de ce nombre est')
  Pour i allant de 0 à 10 par pas de 1 faire
    Afficher(N*i, ' ')
  Fin Pour
Fin
```

Éventuellement, s'il vous reste du temps, leur faire le même algorithme avec un tant que pour leur montrer comment passer de l'un à l'autre.

5. Écrire l'algorithme d'un programme permettant de saisir puis d'afficher une valeur entière comprise entre 1 et 31 inclus ; on recommencera la saisie jusqu'à ce que la valeur soit bien dans les bornes imposées.
   Exemple : valeur saisie : 43 ➔ résultat affiché : valeur non comprise entre 1 et 31 recommencez…
   valeur saisie : 15 ➔ résultat affiché : affichage 15 ok !

```
Début
val : entier
Afficher('Donnez une valeur comprise entre 1 et 31')
Saisir (val)
  Tant que  ((val < 1 ) ou (val > 31)) faire
    Afficher('valeur non comprise entre 1 et 31 recommencez…')
    Saisir(val)
  Fin tant que
  Afficher('affichage ',val,' ok !')
Fin
```

Autre solution (à faire également pour leur montrer la différence entre les deux constructions)

```
Début
val : entier
  Faire
    Afficher('Donnez une valeur comprise entre 1 et 31')
    Saisir (val)
  Tant que ((val < 1 ) ou (val > 31))
  Afficher('affichage ',val,' ok !')
Fin
```