

TP - Programmation

Série de TP N°1 – Structures de contrôle

Rappel :

Structures de contrôle :

1. Structures de contrôle conditionnel

Ces structures sont utilisées pour décider de l'exécution d'un bloc d'instruction : est-ce que ce bloc est exécuté ou non. Ou bien pour choisir entre l'exécution de deux blocs différents. Nous avons deux types de structures conditionnelles :

- a. Test alternatif simple (Si ... FinSi)
- b. Test alternatif double (Si...Sinon...FinSi)

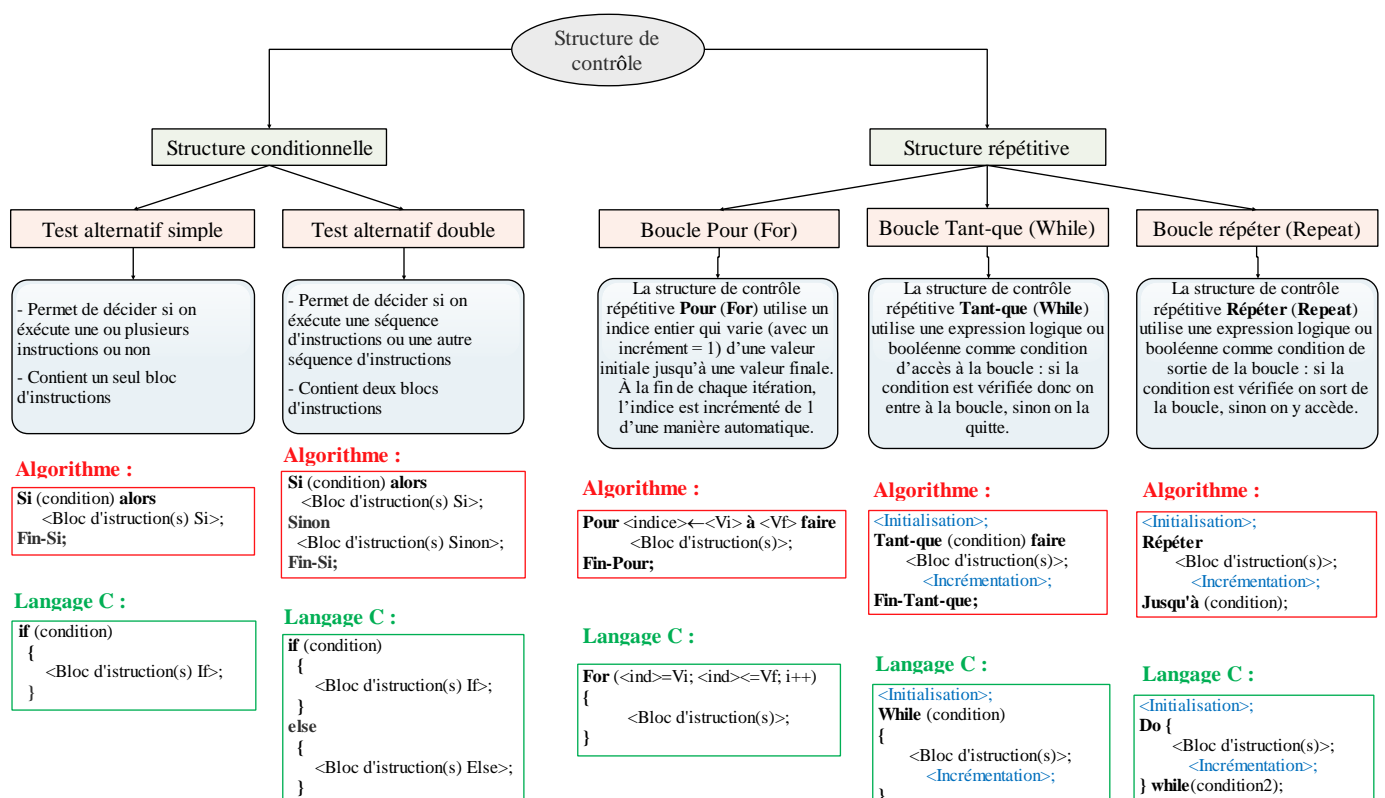
2. Structures de contrôle répétitives

Les structures répétitives nous permettent de répéter un traitement un nombre fini de fois. Par exemple, on veut afficher tous les nombres premiers entre 1 et N (N est un nombre entier positif donné).

Nous avons trois types de structures itératives (boucles) :

- a. Boucle pour (For)
- b. Boucle tant-que (while)
- c. Boucle répéter (repeat)

Les syntaxes des structures de contrôle conditionnelles et de contrôle répétitives sont illustrées dans la figure ci-dessous :



TP - Programmation

Série de TP N°1 – Structures de contrôle

Exercice N°01 : (Algorithme → Programme C)

Soit l'algorithme suivant :

Algorithme Exo1 ;

Variabes

X, P, S : réel ;

i, N : entier ;

Début

Lire (N, X) ;

S ← 0 ;

P ← X ;

i ← 1 ;

Tant-que i ≤ N **faire**

S ← S + P/i ;

P ← P * X ;

i ← i + 1 ;

Fin Tant-que

Écrire (S) ;

Fin.

1. Traduire l'algorithme donné en programme C.
2. Faire le déroulement de l'algorithme donné pour N=2, X= 3 et déduire la valeur de S.
3. Déduire l'expression générale calculée par l'algorithme en fonction de X et N.
4. Réécrire l'algorithme en remplaçant la boucle **Tant que** par la boucle **Pour**.
5. Modifier l'algorithme pour calcule la somme S2 :

$$S2 = X + \frac{X^3}{2} + \frac{X^5}{3} + \frac{X^7}{4} + \dots + N^{\text{ème}} \text{ terme}$$

Exercice N°02 : (Algorithme → Programme C)

Soit l'algorithme suivant :

Algorithme Exo2 ;

Variabes

N, Sn, i : entier;

Début

Lire(N) ;

Sn ← 0 ;

Pour i ← 1 à (N div 2) **faire**

Si (N mod i = 0) **alors**

Sn ← Sn + i;

FinSi

FinPour

Écrire('La somme est : ', Sn);

Fin.

1. Traduire l'algorithme en programme C.
2. Dérouler l'algorithme pour N = 6.
3. Déduire ce que fait l'algorithme.
4. Réécrire le programme en remplaçant la boucle **For** par la boucle **While**.
5. Réécrire l'algorithme pour déterminer si deux nombres entiers N et M sont **amis ou non**.
Deux nombres N et M sont qualifiés d'**amis** si la somme des diviseurs de N (excepté N) est égale à M et la somme des diviseurs de M (excepté M) est égale à N.

Exemple : 220 et 284

TP - Programmation

Série de TP N°1 – Structures de contrôle

Exercice N°03 :

Donner l'organigramme (algorithme) de l'algorithme ci-dessous :

```
Algorithme Vitesse ;  
Variables  
  V, D : réel ;  
  T : entier ;  
Début  
  Lire (D, T) ;  
  V ← D/T ;  
  Si (V <= 80) alors  
    Écrire ('Vitesse normale')  
  Sinon  
    Écrire ('Vitesse élevée') ;  
  FinSi  
Fin.
```

Exercice N°04 :

On demande d'écrire l'algorithme/programme C d'une fiche de paie journalière d'un ouvrier rémunéré à la tâche. Pour cela, on donne :

- La valeur de cette rémunération par pièces réalisées VP,
- Le salaire brut (SB) est calculé selon le nombre de pièces correctes réalisées pendant la journée (NPC) comme suit :

Si $NPC \leq 100$, l'ouvrier touche $NPC * VP$

Si $NPC > 100$, l'ouvrier touche $150 * VP$

- On enlève à la fin 10% du salaire pour les charges sociales (CS).

Calculer et afficher le salaire journalier brut (SB), les charges sociales (CS) et salaire journalier net (SN).

NB : Salaire brut=salaire totale ; Salaire net=salaire sans les charges sociales.