

TP - Programmation

Corrigé de la 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)

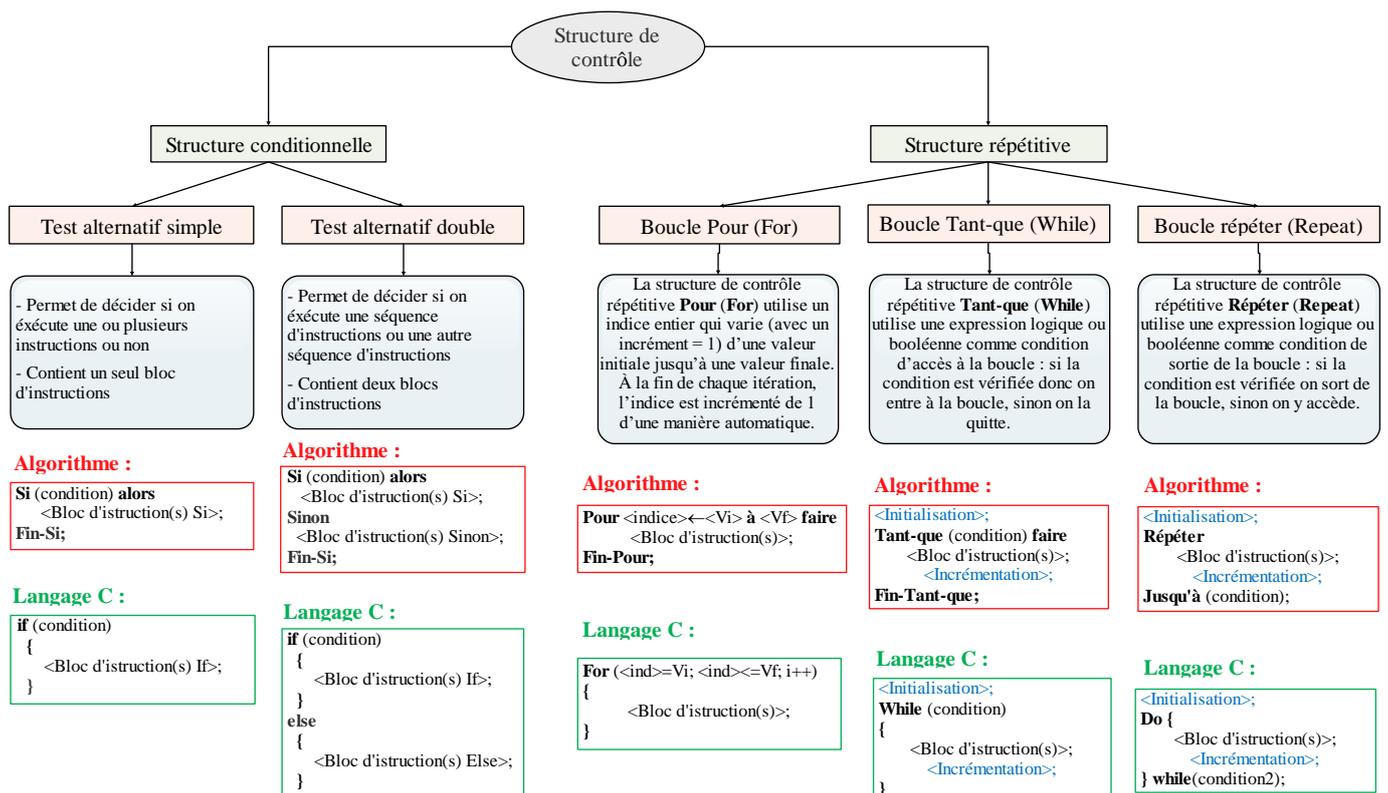
2. Structures de contrôle répétitives

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 :



Exercice N°01 : (Algorithme → Programme C)

Soit l'algorithme suivant :

Algorithme Ex01 ;**Variables**

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 :

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

Solution :

1. Traduire l'algorithme donné en programme C

Algorithme	Programme C
Algorithme Ex01 ; Variables 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 Ecrire (S) ; Fin.	<pre>#include<stdio.h> int main() { float X, P, S; int i,N; /* Entrées */ printf("Donner la valeur de N et X : "); scanf("%d %f", &N, &X); /* Traitements */ S=0; P=X; i=1; while (i<=N) { S=S+P/i; P=P*X; i=i+1; } /* Sorties */ printf("S= %f",S); return 0; }</pre>

2. Faire le déroulement de l'algorithme donné pour N=2, X= 3 et déduire la valeur de S

Instructions	Variables					Affichage
	N	X	i	P	S	
Lire (N,X)	2	3	/	/	/	/
S ← 0;	2	3	/	/	0	/
P ← X;	2	3	/	3	0	/
i ← 1;	2	3	1	3	0	/
Tant-que i <= N faire (1<=2) Vrai						/
S ← S+ P/i ;	2	3	1	3	3	/
P ← P*X ;	2	3	1	9	3	/
i ← i+1 ;	2	3	2	9	3	/
Tant-que i <= N faire (2<=2) Vrai						/
S ← S+ P/i ;	2	3	2	9	7.5	/
P ← P*X ;	2	3	2	27	7.5	/
i ← i+1 ;	2	3	3	27	7.5	/
Tant-que i <= N faire (3<=2) Faux On sort de la boucle						
Ecrire (S) ;	2	3	3	27	7.5	7.5

3. Déduire l'expression générale calculée par l'algorithme en fonction de X et N

Selon le déroulement ci-dessus, nous avons :

Pour i = 1, nous avons S = 3.

Pour i = 2, nous avons $S = 3 + \frac{9}{2} = 7.5$.

Pour i = N nous aurons : $x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^N}{N}$

On peut généraliser par la formule suivante :

$$S = \sum_{i=1}^N \frac{x^i}{i} \text{ ou } S = x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^N}{N}$$

4. Réécrire l'algorithme en remplaçant la boucle Tant que par la boucle Pour

Algorithme avec la boucle Tantque	Algorithme avec la boucle pour
<p>Algorithme Exo1 ;</p> <p>Variables X, P, S : réel ; i, N : entier ;</p> <p>Début Lire (N, X) ; S ← 0 ; P ← X ; i ← 1 ;</p> <p>Tant-que i ≤ N faire S ← S+ P/i ; P ← P*X ; i ← i+1 ;</p> <p>Fin Tant-que Ecrire (S) ;</p> <p>Fin.</p>	<p>Algorithme Exo1 ;</p> <p>Variables X, P, S : réel ; i, N : entier ;</p> <p>Début Lire (N, X) ; S ← 0 ; P ← X ;</p> <p>Pour i ← 1 à N faire S ← S+ P/i ; P ← P*X ;</p> <p>FinPour Ecrire (S) ;</p> <p>Fin.</p>

5. Modifier l'algorithme pour calculer la somme S2 :

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

Algorithme avec la boucle while
<p>Algorithme Exo1 ;</p> <p>Variables X, P, S : réel ; i, N : entier ;</p> <p>Début Lire (N, X) ; S ← 0 ; P ← X ; i ← 1 ;</p> <p>Tant-que i ≤ N faire S ← S+ P/i ; P ← P*X*X ; i ← i+1 ;</p> <p>Fin Tant-que Ecrire (S) ;</p> <p>Fin.</p>

Exercice N°02 : (Algorithmes → Programme C)

Soit l'algorithme suivant :

<p>Algorithme Exo2 ;</p> <p>Variables N, Sn, i : entier;</p> <p>Début Lire(N) ; Sn ← 0 ; Pour i ← 1 à (N div 2) faire Si (N mod i = 0) alors Sn ← Sn + i; FinSi FinPour Ecrire('La somme est : ', Sn); Fin.</p>	<ol style="list-style-type: none">1. Traduire l'algorithme en programme C.2. Dérouler l'algorithme pour N = 6.3. Dédire 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. <p>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.</p> <p>Exemple : 220 et 284</p>
--	--

Solution :

1. Traduire l'algorithme en programme C

Algorithme	Programme C
<p>Algorithme Exo2 ;</p> <p>Variables N, Sn, i : entier;</p> <p>Début Lire(N) ; Sn ← 0 ; Pour i ← 1 à (N div 2) faire Si (N mod i = 0) alors Sn ← Sn + i; FinSi FinPour Ecrire('La somme est : ', Sn); Fin.</p>	<pre>#include<stdio.h> int main() { int N,Sn,i; /* Entrées */ printf("Donner la Valeur de N :"); scanf("%d", &N); /* Traitements */ Sn=0; for (i=1;i<=N/2;i++) { if(N % i == 0) { Sn=Sn+i; } } /* Sorties */ printf("La somme est : %d ",Sn); return 0; }</pre>

2. Dérouler l'algorithme pour N=6

Instructions	Variables			Affichage
	N	i	Sn	
Lire(N)	6	/	/	
Sn ← 0	6	/	0	
Pour i ← 1 Si (6 mod 1 = 0) 0 = 0 Vrai Sn ← Sn + i Sn ← 0 + 1 Sn ← 1	6	1	1	
Pour i ← 2 Si (6 mod 2 = 0) 0 = 0 Vrai Sn ← Sn + i Sn ← 1 + 2 Sn ← 3	6	2	3	
Pour i ← 3 Si (6 mod 3 = 0) 0 = 0 Vrai Sn ← Sn + i Sn ← 3 + 3 Sn ← 6	6	3	6	
Ecrire(' La somme est : ', Sn)	6	3	6	La somme est : 6

3. Déduire ce que fait l'algorithme

L'algorithme permet de calculer et afficher la somme des diviseurs d'un nombre entier N excepté lui-même (ou sans lui-même, sans compter lui-même, ou encore, sans N).

4. Réécrire le programme en remplaçant la boucle For par la boucle While

Programme C avec la boucle for	Programme C avec la boucle while
<pre>#include<stdio.h> int main() { int N,Sn,i; /* Entrées */ printf("Donner la Valeur de N :"); scanf("%d", &N); /* Traitements */ Sn=0; for (i=1;i<=N/2;i++) { if(N % i == 0) { Sn=Sn+i; } } /* Sorties */ printf("La somme est : %d ",Sn); return 0; }</pre>	<pre>#include<stdio.h> int main() { int N,Sn,i; /* Entrées */ printf("Donner la Valeur de N :"); scanf("%d", &N); /* Traitements */ Sn=0; i=1 ; while(i<=N/2) { if(N % i == 0) { Sn=Sn+i; } i=i+1 ; } /* Sorties */ printf("La somme est : %d ",Sn); return 0; }</pre>

5. Réécrire l'algorithme pour déterminer si deux nombres entiers N et M sont amis ou non

Algorithme « NombresAmis »

Algorithme NombresAmis ;

Var

N, Sn, i : entier ;

M, Sm : entier ;

Début

Lire(N) ;

Lire(M) ;

Sn ← 0 ;

Pour i ← 1 à (N div 2) faire

 Si (N mod i = 0) alors

 Sn ← Sn + i;

 Finsi;

Finpour;

Sm ← 0 ;

Pour i ← 1 à (M div 2) faire

 Si (M mod i = 0) alors

 Sm ← Sm + i;

 FinSi;

FinPour;

Si (Sn=M) et (Sm=N) alors

 Ecrire (N, ' et ', M, ' sont amis')

Sinon

 Ecrire (N, ' et ', M, ' ne sont pas amis') ;

Finsi ;

Fin.

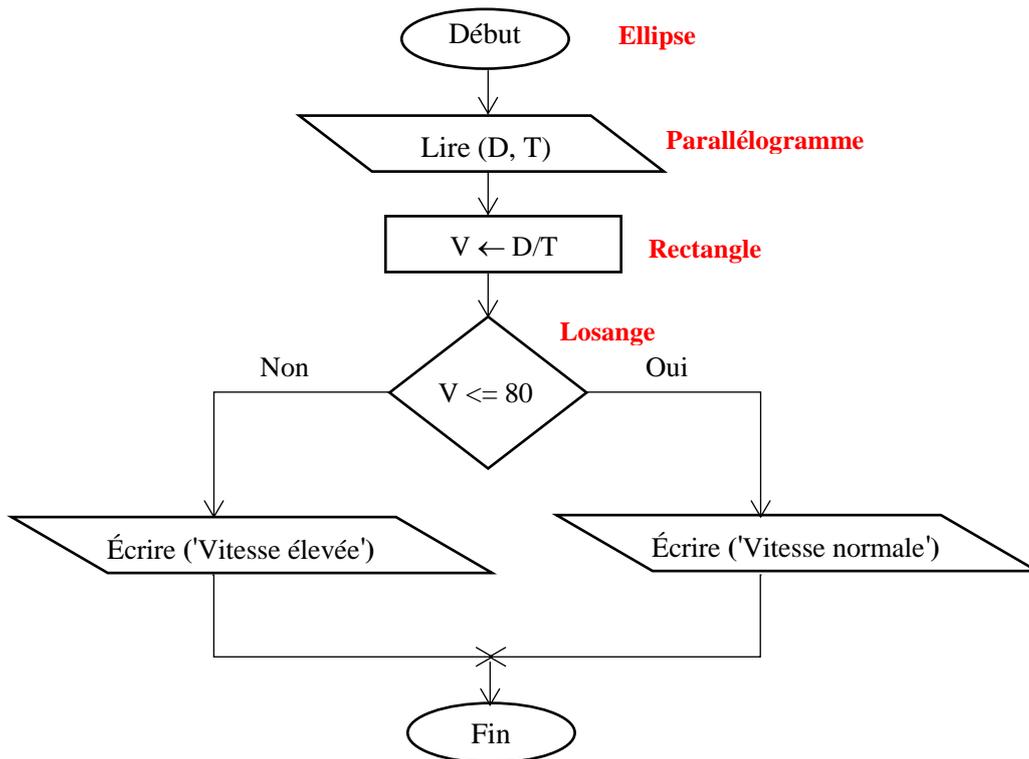
Exercice N°03 :

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.
```

Solution :

Organigramme (algorithme) :



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.

Solution :

Algorithme	Programme C
<p>Algorithme Exo3;</p> <p>Variables NPC,VP, SB, SN, CS : réel ;</p> <p>Début</p> <p><i>/* Entrées */</i> Lire (NPC,VP) ;</p> <p><i>/* Traitements */</i> Si (NPC <= 100) Alors SB ← NPC * VP;</p> <p>Sinon SB ← VP*150 ;</p> <p>Fin-Si CS ← SB * 0.1; <i>//ou CS ← SB*10 /100;</i> SN ← SB – CS;</p> <p><i>/* Sorties */</i> Ecrire ('Le salaire brut =', SB) ; Ecrire ('Les charges sociales =', CS) ; Ecrire ('Le salaire net =', SN) ;</p> <p>Fin.</p>	<pre>#include<stdio.h> int main() { float NPC,VP,SB,SN,CS; <i>/* Entrées */</i> scanf("%f %f", &NPC, &VP); <i>/* Traitements */</i> if (NPC<=100) { SB=NPC*VP; } else SB=VP*150; CS=SB*0.1; SN=SB-CS; <i>/* Sorties */</i> printf("Le salaire brut = %f \n", SB); printf("Les charges sociales = %f \n", CS); printf("Le salaire net = %.2f", SN); return 0; }</pre>