

TP Structure des ordinateurs et applications

Corrigé de la série de TP N°5

Rappel :

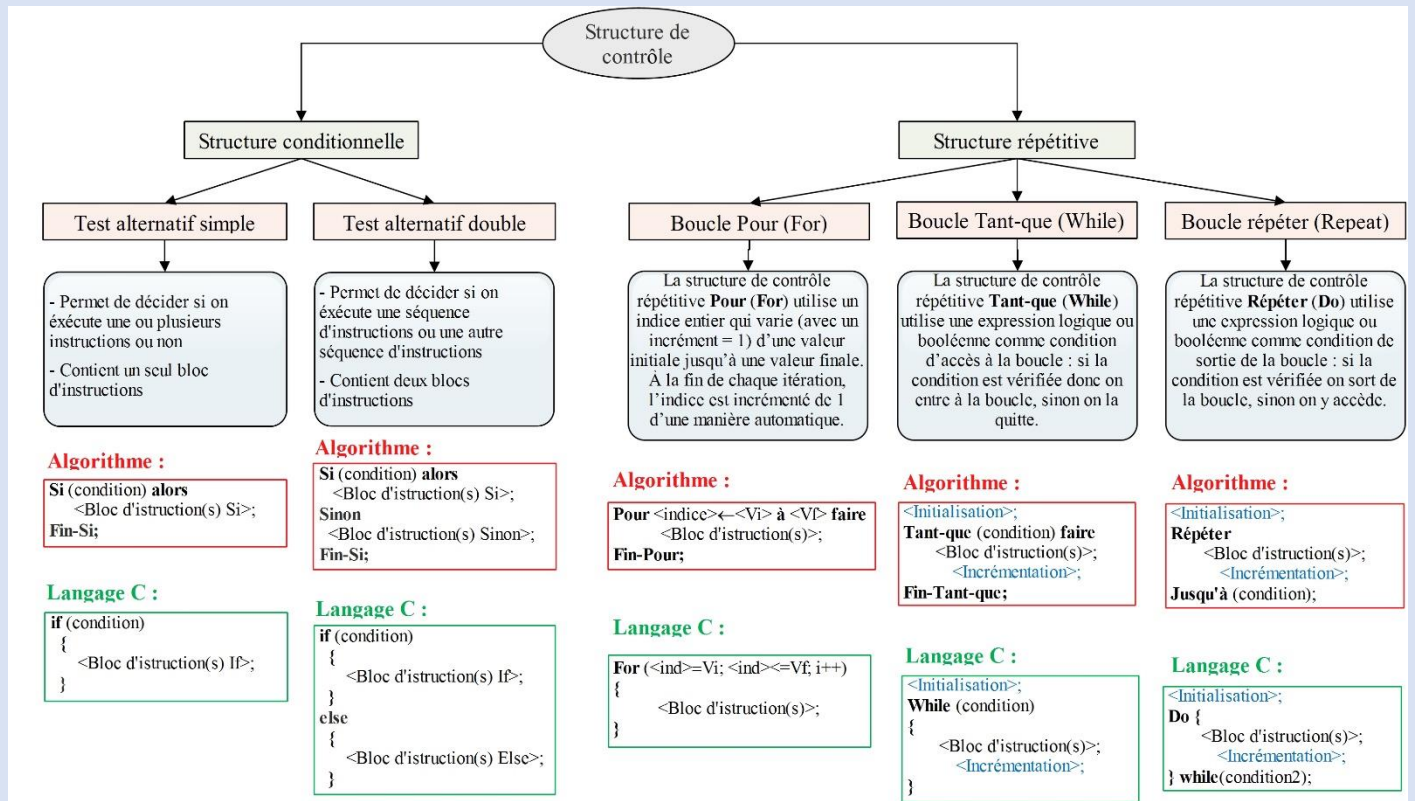
Structures de contrôle répétitives

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) :

1. Boucle pour (For)
2. Boucle tant-que (while)
3. Boucle répéter (repeat)

Les syntaxes des trois boucles sont illustrées dans la figure ci-dessous :



La différence entre la boucle **répéter** et la boucle **tant-que** est :

- La condition de **répéter** est toujours l'inverse de la condition **tant-que** : pour **répéter** c'est la condition de sortie de la boucle, et pour **tant-que** c'est la condition d'entrer.
- Le teste de la condition est à la fin de la boucle (la fin de l'itération) pour **répéter**. Par contre, il est au début de l'itération pour la boucle **tant-que**. C'est-à-dire, dans **tant-que** on teste la condition avant d'entrer à l'itération, et dans **répéter** on fait l'itération après on teste la condition.

Exercice N°01 : (Algorithme → Programme en langage C)

1) Traduire l'algorithme en Programme C

Algorithme	Programme C
<p>Algorithme Exo1 ;</p> <p>Variables</p> <p>X, P, S : réel ;</p> <p>i, N : entier ;</p> <p>Début</p> <p><i>// Entrées</i></p> <p>Ecrire ("Donner les valeurs de N et X : ") ;</p> <p>Lire (N, X) ;</p> <p><i>// Traitement</i></p> <p>S ← 0 ;</p> <p>P ← X ;</p> <p>Pour i ← 1 à N faire</p> <p> S ← S+ P/i ;</p> <p> P ← P*X ;</p> <p>Fin-Pour</p> <p><i>// Sortie</i></p> <p>Ecrire ("S = ", S) ;</p> <p>Fin.</p>	<pre>#include<stdio.h> int main() { float X,P,S; int i,N; // Entrées printf("Donner les valeurs de N et X \n"); scanf("%d %f", &N,&X); // Traitement S=0; P=X; for (i=1;i<=N;i++) { S=S+P/i; P=P*X; } // Sortie printf("S=%.3f",S); }</pre>

2) Compiler et exécuter le programme pour N=3, X=3.

The screenshot shows the Code::Blocks IDE with a C program in the editor and its execution output in a terminal window. The program code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     float X,P,S;
5     int i,N;
6     // Entrées
7     printf("Donner les valeurs de N et X \n");
8     scanf("%d %f", &N,&X);
9
10
11     // Traitement
12     S=0;
13     P=X;
14     for (i=1;i<=N;i++)
15     {
16         S=S+P/i;
17         P=P*X;
18     }
19
20     // Sortie
21     printf("S=%.3f",S);
22 }
```

The terminal window shows the following output:

```
C:\Users\Ahmed\Desktop\TP4 >
Donner les valeurs de N et X
3 3
S=16.500
Process returned 0 (0x0)   execution time : 2.988 s
Press any key to continue.
```

An arrow points from the terminal window to a box labeled "Après l'exécution" (After execution).

3) Dérouler le programme pour N=3, X=3.

Instructions	Variables					Affichage
	N	X	i	P	S	
Lire (N,X);	3	3	/	/	/	/
S ← 0;	3	3	/	/	0	/
P ← X; P ← 3	3	3	/	3	0	/
Pour i=1 alors S ← S+ P/i ; S ← 0+ 3/1 ; S ← 3 ; P ← P*X ; P ← 3*3 ; P ← 9 ; Fin-Pour	3	3	1		3	
Pour i=2 alors S ← S+ P/i ; S ← 0+ 3/1+ 9/2 ; S ← 7.5 ; P ← P*X ; P ← 3*3*3 ; P ← 27 ; Fin-Pour	3	3	1		7.5	
Pour i=3 alors S ← S+ P/i ; S ← 0+ 3/1+ 9/2 + 27/3 ; S ← 16.5 ; P ← P*X ; P ← 3*3*3*3 ; P ← 81 ; Fin-Pour	3	3	1		16.5	
Ecrire ('S = ', S) ;	3	3	1	81	16.5	S = 16.500

4) Dédurre l'expression générale du résultat S 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 = 3, nous avons $S = 3 + \frac{9}{2} + \frac{27}{3} = 16.5$

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

On peut généraliser par la formule suivante :

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

5) Réécrire l'algorithme en remplaçant la boucle *Pour* par la boucle *Tant-que*.

Algorithme	Programme C
<p>Algorithme Ex01 ;</p> <p>Variables X, P, S : réel ; i, N : entier ;</p> <p>Début // Entrées</p> <p>Ecrire ("Donner les valeurs de N et X : ") ;</p> <p>Lire (N, X) ;</p> <p>// Traitement</p> <p>S ← 0 ; P ← X ; i ← 1 ;</p> <p>Tant-que i ≤ N faire</p> <p style="padding-left: 20px;">S ← S+ P/i ; P ← P*X ; i ← i+1 ;</p> <p>Fin Tant-que</p> <p>// Sortie</p> <p>Ecrire ("S =", S) ;</p> <p>Fin.</p>	<pre>#include<stdio.h> int main() { float X,P,S; int i,N; // Entrées printf("Donner les valeurs de N et X \n"); scanf("%d %f", &N,&X); // Traitement S=0; P=X; i=1; while (i<=N) { S=S+P/i; P=P*X; i=i+1; } // Sortie printf("S=%f",S); }</pre>

6) Réécrire l'algorithme en remplaçant la boucle *Pour* par la boucle *Répéter*.

Algorithme	Programme C
<p>Algorithme Ex01 ;</p> <p>Variables X, P, S : réel ; i, N : entier ;</p> <p>Début // Entrées</p> <p>Ecrire ("Donner les valeurs de N et X : ") ;</p> <p>Lire (N, X) ;</p> <p>// Traitement</p> <p>S ← 0 ; P ← X ; i ← 1 ;</p> <p>Répéter</p> <p style="padding-left: 20px;">S ← S+ P/i ; P ← P*X ; i ← i+1 ;</p> <p>Jusqu'a i > N;</p> <p>// Sortie</p> <p>Ecrire ("S =", S) ;</p> <p>Fin.</p>	<pre>#include<stdio.h> int main() { float X,P,S; int i,N; // Entrées printf("Donner les valeurs de N et X \n"); scanf("%d %f", &N,&X); // Traitement S=0; P=X; i=1; do { S=S+P/i; P=P*X; i=i+1; } while(i<=N); // Sortie printf("S=%f",S); }</pre>

7) 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	Programme C
<p>Algorithme Exo2 ;</p> <p>Variables X, P, S : réel ; i, N : entier ;</p> <p>Début <i>// Entrées</i> Ecrire ("Donner les valeurs de N et X : ") ; Lire (N, X) ; <i>// Traitement</i> S ← 0 ; P ← X ; Pour i:=1 à N faire S ← S+ P/i ; P ← P*X*X ; Fin-Pour ; <i>// Sortie</i> Ecrire ("S =", S) ; Fin.</p>	<pre>#include<stdio.h> int main() { float X,P,S; int i,N; // Entrées printf("Donner les valeurs de N et X \n"); scanf("%d %f", &N,&X); // Traitement S=0; P=X; for (i=1;i<=N;i++) { S=S+P/i; P=P*X*X; } // Sortie printf("S=% .3f",S); }</pre>

Exercice N°02 : (Enoncé du problème → Algorithme → Programme en langage C)

1) Calculer la somme $S = 1^2 + 3^2 + 5^2 + \dots + (2 * N + 1)^2$

Algorithme	Programme C
<p>Algorithme exo2_1;</p> <p>variables S, N, i: entier;</p> <p>Début <i>// Entrées</i> Ecrire("Donner la valeur de N : "); Lire(N); <i>// Traitement</i> S ← 0; Pour i ← 0 à N faire S ← S + (2*i+1)²; Fin-pour <i>// Sortie</i> Ecrire ("S =", S); Fin.</p>	<pre>#include<stdio.h> int main() { int S,N,i; // Entrées printf("Donner la valeur de N \n"); scanf("%d", &N); // Traitement S=0; for (i=0;i<=N;i++) S=S+pow(2*i+1,2); // Sortie printf("S=%d",S); }</pre>

2) Calculer le produit $P = 1 \times 2 \times 3 \times \dots \times N$

Algorithme	Programme C
<p>Algorithme exo2_2;</p> <p>Variables</p> <p> N, i, P: entier ;</p> <p>Début</p> <p> // Entrées</p> <p> Ecrire("Donner la valeur de N : ");</p> <p> Lire(N);</p> <p> // Traitement</p> <p> P ← 1;</p> <p> Pour i ← 1 à N faire</p> <p> P ← P* i;</p> <p> Fin-pour;</p> <p> // Sortie</p> <p> Ecrire("P =", P);</p> <p>Fin.</p>	<pre>#include<stdio.h> int main() { int N,i,P; // Entrées printf("Donner la valeur de N \n"); scanf("%d", &N); // Traitement P=1; for (i=1;i<=N;i++) P=P*i; // Sortie printf("P=%d",P); }</pre>

3) Calculer la somme $S = X + \frac{X^3}{2!} + \frac{X^5}{4!} + \frac{X^7}{6!} + \dots$ ($N^{\text{ème}}$ terme)

Algorithme	Programme C
<p>Algorithme exo2_4;</p> <p>Variables</p> <p>N, i, F : entier;</p> <p>X, P, S : réel;</p> <p>Début</p> <p><i>// Entrées</i></p> <p>Ecrire("Donner la valeur de N : ");</p> <p>Lire(N);</p> <p>Ecrire("Donner la valeur de X : ");</p> <p>Lire(X);</p> <p><i>// Traitement</i></p> <p>S ← 0;</p> <p>P ← X;</p> <p>F ← 1;</p> <p>Pour i:=0 à (N-1) faire</p> <p> S ← S + P/F;</p> <p> P ← P*X*X;</p> <p> F ← F*(2*i+1)*(2*i+2);</p> <p>Fin-Pour;</p> <p><i>// Sortie</i></p> <p>Ecrire("S = ", S);</p> <p>Fin.</p>	<pre>#include<stdio.h> int main() { int N,i,F; float X,P,S; // Entrées printf("Donner la valeur de N \n"); scanf("%d", &N); printf("Donner la valeur de X \n"); scanf("%f", &X); // Traitement S=0; P=X; F=1; for (i=0;i<=N-1;i++) { S=S+P/F; P=P*X*X; F=F*(2*i+1)*(2*i+2); } // Sortie printf("S=%.3f",S); }</pre>

4) Calculer la somme $S = X - X^2 + X^3 - \dots \pm X^N$

Algorithme	Programme C
<p>Algorithme exo2_5;</p> <p>Variables</p> <p> N, i : entier;</p> <p> X, P, S : réel;</p> <p>Début</p> <p> // Entrées</p> <p> Ecrire("Donner la valeur de N : ");</p> <p> Lire(N);</p> <p> Ecrire("Donner la valeur de X : ");</p> <p> Lire(X);</p> <p> // Traitement</p> <p> S ← 0;</p> <p> P ← -1;</p> <p> Pour i ← 1 à N faire</p> <p> P ← -P*X;</p> <p> S ← S+P;</p> <p> Fin-Pour;</p> <p> // Sortie</p> <p> Ecrire("S= ", S);</p> <p>Fin.</p>	<pre>#include<stdio.h> int main() { int N,i; float X,P,S; // Entrées printf("Donner la valeur de N : \n"); scanf("%d", &N); printf("Donner la valeur de X : \n"); scanf("%f", &X); // Traitement S=0; P=-1; for (i=1;i<=N;i++) { P=-P*X; S=S+P; } // Sortie printf("S=% .3f",S); }</pre>

Exercice N°03 :

Soit A et B deux entiers tel-que $A < B$. Introduire N valeurs entières entre A et B, et réaliser la somme de valeurs pairs non-nuls et le produit des valeurs impaires.

Algorithme	Programme C
<pre>Algorithme Exo2_7; Variables A, B, N, i, Vi, S, P : entier; Début // Entrées Répéter Ecrire("Donner la valeur de A et B : "); Lire(A, B); Jusqu'à (A<B); Ecrire("Donner la valeur de N : "); Lire(N); S ← 0; P ← 1; Pour i ← 1 à N faire Répéter Lire(Vi) Jusqu'à (Vi>=A) ET (Vi<=B); Si Vi mod 2 = 0 Alors S ← S + Vi; Sinon P ← P * Vi; Fin-Si; Fin-Pour; // Sortie Ecrire("S = ", S, " P = ", P); Fin.</pre>	<pre>#include<stdio.h> int main() { int A, B, N, i, Vi, S, P; // Entrées do{ printf("Donner la valeur de A et B : \n"); scanf("%d %d", &A, &B); } while(A>B); printf("Donner la valeur de N : \n"); scanf("%d", &N); S=0; P=1; for (i=1;i<=N;i++) { do{ scanf("%d", &Vi); } while (Vi<A Vi>B); if (Vi % 2==0) S=S + Vi; else P=P * Vi; } // Sortie printf("S = %d et P= %d", S,P); }</pre>