

## TP Structure des ordinateurs et applications

### Corrigé de la série de TP N°5

#### Rappel :

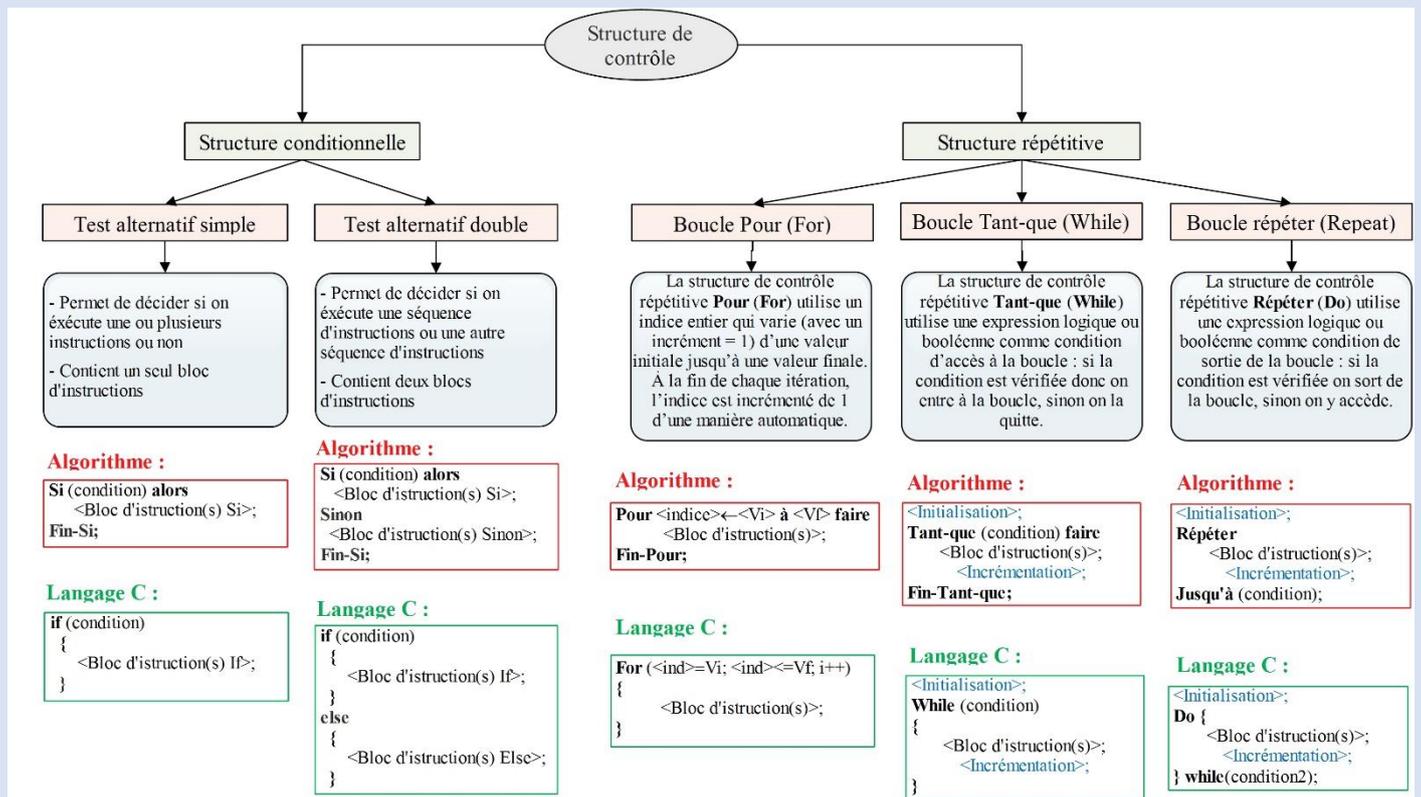
#### Structures de contrôle répétitives

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><b>Algorithme</b> Exo1 ;</p> <p><b>Variables</b></p> <p>X, P, S : réel ;</p> <p>i, N : entier ;</p> <p><b>Début</b></p> <p><i>// Entrées</i></p> <p><b>Ecrire</b> ("Donner les valeurs de N et X : ") ;</p> <p><b>Lire</b> (N, X) ;</p> <p><i>// Traitement</i></p> <p>S ← 0 ;</p> <p>P ← X ;</p> <p><b>Pour</b> i ← 1 à N <b>faire</b></p> <p>    S ← S+ P/i ;</p> <p>    P ← P*X ;</p> <p><b>Fin-Pour</b></p> <p><i>// Sortie</i></p> <p><b>Ecrire</b> ("S = ", S) ;</p> <p><b>Fin.</b></p>	<pre>#include&lt;stdio.h&gt; int main() {     float X,P,S;     int i,N;     // Entrées     printf("Donner les valeurs de N et X \n");     scanf("%d %f", &amp;N,&amp;X);      // Traitement     S=0;     P=X;     for (i=1;i&lt;=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	/
<b>Pour i=1 alors</b> S ← S+ P/i ; S ← 0+ 3/1 ; S ← 3 ; P ← P*X ; P ← 3*3 ; P ← 9 ; <b>Fin-Pour</b>	3	3	1		3	
<b>Pour i=2 alors</b> S ← S+ P/i ; S ← 0+ 3/1+ 9/2 ; S ← 7.5 ; P ← P*X ; P ← 3*3*3 ; P ← 27 ; <b>Fin-Pour</b>	3	3	1		7.5	
<b>Pour i=3 alors</b> 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 ; <b>Fin-Pour</b>	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><b>Algorithme</b> Ex01 ;</p> <p><b>Variables</b>  X, P, S : réel ;  i, N : entier ;</p> <p><b>Début</b>  // Entrées</p> <p><b>Ecrire</b> ("Donner les valeurs de N et X : ") ;</p> <p><b>Lire</b> (N, X) ;</p> <p>// Traitement</p> <p>S ← 0 ;  P ← X ;  i ← 1 ;</p> <p><b>Tant-que</b> i ≤ N faire</p> <p style="padding-left: 20px;">S ← S+ P/i ;  P ← P*X ;  i ← i+1 ;</p> <p><b>Fin Tant-que</b></p> <p>// Sortie</p> <p><b>Ecrire</b> ("S =", S) ;</p> <p><b>Fin.</b></p>	<pre>#include&lt;stdio.h&gt; int main() {     float X,P,S;     int i,N;     // Entrées     printf("Donner les valeurs de N et X \n");     scanf("%d %f", &amp;N,&amp;X);     // Traitement     S=0;     P=X;     i=1;     while (i&lt;=N)     {         S=S+P/i;         P=P*X;         i=i+1;     }     // Sortie     printf("S=%0.3f",S); }</pre>

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

Algorithme	Programme C
<p><b>Algorithme</b> Ex01 ;</p> <p><b>Variables</b>  X, P, S : réel ;  i, N : entier ;</p> <p><b>Début</b>  // Entrées</p> <p><b>Ecrire</b> ("Donner les valeurs de N et X : ") ;</p> <p><b>Lire</b> (N, X) ;</p> <p>// Traitement</p> <p>S ← 0 ;  P ← X ;  i ← 1 ;</p> <p><b>Répéter</b></p> <p style="padding-left: 20px;">S ← S+ P/i ;  P ← P*X ;  i ← i+1 ;</p> <p><b>Jusqu'a</b> i &gt; N;</p> <p>// Sortie</p> <p><b>Ecrire</b> ("S =", S) ;</p> <p><b>Fin.</b></p>	<pre>#include&lt;stdio.h&gt; int main() {     float X,P,S;     int i,N;     // Entrées     printf("Donner les valeurs de N et X \n");     scanf("%d %f", &amp;N,&amp;X);     // Traitement     S=0;     P=X;     i=1;     do     {         S=S+P/i;         P=P*X;         i=i+1;     }     while(i&lt;=N);     // Sortie     printf("S=%0.3f",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><b>Algorithme</b> Exo2 ;</p> <p><b>Variables</b>  X, P, S : réel ;  i, N : entier ;</p> <p><b>Début</b>  // Entrées  <b>Ecrire</b> ("Donner les valeurs de N et X : ") ;  <b>Lire</b> (N, X) ;  // Traitement  S ← 0 ;  P ← X ;  <b>Pour</b> i:=1 à N faire      S ← S+ P/i ;      P ← P*X*X ;  <b>Fin-Pour</b> ;  // Sortie  <b>Ecrire</b> ("S =", S) ;  <b>Fin.</b></p>	<pre>#include&lt;stdio.h&gt; int main() {     float X,P,S;     int i,N;     // Entrées     printf("Donner les valeurs de N et X \n");     scanf("%d %f", &amp;N,&amp;X);      // Traitement     S=0;     P=X;     for (i=1;i&lt;=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><b>Algorithme</b> exo2_1;</p> <p><b>variables</b>  S, N, i: entier;</p> <p><b>Début</b>  // Entrées  <b>Ecrire</b>("Donner la valeur de N : ");  <b>Lire</b>(N);  // Traitement  S ← 0;  <b>Pour</b> i ← 0 à N faire      S ← S + (2*i+1)<sup>2</sup>;  <b>Fin-pour</b>  // Sortie  <b>Ecrire</b> ("S =", S);  <b>Fin.</b></p>	<pre>#include&lt;stdio.h&gt; int main() {     int S,N,i;     // Entrées     printf("Donner la valeur de N \n");     scanf("%d", &amp;N);      // Traitement     S=0;     for (i=0;i&lt;=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><b>Algorithme</b> exo2_2;</p> <p><b>Variables</b></p> <p>    N, i, P: entier ;</p> <p><b>Début</b></p> <p>    // Entrées</p> <p>    <b>Ecrire</b>("Donner la valeur de N : ");</p> <p>    <b>Lire</b>(N);</p> <p>    // Traitement</p> <p>    P ← 1;</p> <p>    <b>Pour</b> i ← 1 à N <b>faire</b></p> <p>        P ← P* i;</p> <p>    <b>Fin-pour</b>;</p> <p>    // Sortie</p> <p>    <b>Ecrire</b>("P =", P);</p> <p><b>Fin.</b></p>	<pre>#include&lt;stdio.h&gt; int main() {     int N,i,P;     // Entrées     printf("Donner la valeur de N \n");     scanf("%d", &amp;N);      // Traitement     P=1;     for (i=1;i&lt;=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><b>Algorithme</b> exo2_4;</p> <p><b>Variables</b></p> <p>N, i, F : entier;</p> <p>X, P, S : réel;</p> <p><b>Début</b></p> <p><i>// Entrées</i></p> <p><b>Ecrire</b>("Donner la valeur de N : ");</p> <p><b>Lire</b>(N);</p> <p><b>Ecrire</b>("Donner la valeur de X : ");</p> <p><b>Lire</b>(X);</p> <p><i>// Traitement</i></p> <p>S ← 0;</p> <p>P ← X;</p> <p>F ← 1;</p> <p><b>Pour</b> i:=0 à (N-1) <b>faire</b></p> <p style="padding-left: 20px;">S ← S + P/F;</p> <p style="padding-left: 20px;">P ← P*X*X;</p> <p style="padding-left: 20px;">F ← F*(2*i+1)*(2*i+2);</p> <p><b>Fin-Pour</b>;</p> <p><i>// Sortie</i></p> <p><b>Ecrire</b>("S = ", S);</p> <p><b>Fin.</b></p>	<pre>#include&lt;stdio.h&gt; int main() {     int N,i,F;     float X,P,S;     // Entrées     printf("Donner la valeur de N \n");     scanf("%d", &amp;N);     printf("Donner la valeur de X \n");     scanf("%f", &amp;X);      // Traitement     S=0;     P=X;     F=1;     for (i=0;i&lt;=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><b>Algorithme</b> exo2_5;</p> <p><b>Variables</b></p> <p style="padding-left: 20px;">N, i : entier;</p> <p style="padding-left: 20px;">X, P, S : réel;</p> <p><b>Début</b></p> <p style="padding-left: 20px;"><i>// Entrées</i></p> <p style="padding-left: 20px;">Ecrire("Donner la valeur de N : ");</p> <p style="padding-left: 20px;">Lire(N);</p> <p style="padding-left: 20px;">Ecrire("Donner la valeur de X : ");</p> <p style="padding-left: 20px;">Lire(X);</p> <p style="padding-left: 20px;"><i>// Traitement</i></p> <p style="padding-left: 20px;">S ← 0;</p> <p style="padding-left: 20px;">P ← -1;</p> <p style="padding-left: 20px;"><b>Pour</b> i ← 1 à N <b>faire</b></p> <p style="padding-left: 40px;">P ← -P*X;</p> <p style="padding-left: 40px;">S ← S+P;</p> <p style="padding-left: 20px;"><b>Fin-Pour</b>;</p> <p style="padding-left: 20px;"><i>// Sortie</i></p> <p style="padding-left: 20px;">Ecrire("S= ", S);</p> <p><b>Fin.</b></p>	<pre>#include&lt;stdio.h&gt; int main() {     int N,i;     float X,P,S;     // Entrées     printf("Donner la valeur de N : \n");     scanf("%d", &amp;N);     printf("Donner la valeur de X : \n");     scanf("%f", &amp;X);      // Traitement     S=0;     P=-1;     for (i=1;i&lt;=N;i++)     {         P=-P*X;         S=S+P;     }     // Sortie     printf("S=% .3f",S); }</pre>

### Exercice N°03 :

Soit A et B deux entier 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&lt;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&gt;=A) ET (Vi&lt;=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&lt;stdio.h&gt; int main() {     int A, B, N, i, Vi, S, P;     // Entrées     do{         printf("Donner la valeur de A et B : \n");         scanf("%d %d", &amp;A, &amp;B);     } while(A&gt;B);     printf("Donner la valeur de N : \n");     scanf("%d", &amp;N);     S=0;     P=1;     for (i=1;i&lt;=N;i++)     {         do{             scanf("%d", &amp;Vi);         } while (Vi&lt;A    Vi&gt;B);         if (Vi % 2==0)             S=S + Vi;         else             P=P * Vi;     }     // Sortie     printf("S = %d et P= %d", S,P); }</pre>