Le corps de programme

Le corps d'un algorithme est constitué d'un ensemble d'**actions / instructions** ordonnées de manière **séquentielle et logique**. Ces instructions se divisent en **cinq types distincts** :

1 *Lecture*

Cette opération consiste à introduire des données dans l'algorithme. Une lecture consiste à donner une valeur arbitraire à une variable.

2 *Écriture*

Cette opération implique l'affichage de données. Elle permet d'afficher des résultats ou des messages.

3 *Affectation*

Elle permet de modifier les valeurs des variables en leur assignant de nouvelles valeurs.

Structures de contrôle

Ces structures permettent de modifier la séquentialité de l'algorithme. Elles sont utilisées pour sélectionner différents chemins d'exécution ou pour répéter un traitement.

4 *Structure de* **Test alternatif simple / double / imbriqué**

5 *Structures* **répétitives (itérative)**

- *La boucle* **Pour**
- *La boucle* **Tantque**
- *La boucle* **Répéter**

Structures de contrôle répétitives

↻ Les structures répétitives nous permettent de **répéter un traitement un nombre fini de fois**.

Exemple : afficher tous les nombres premiers entre 1 et N (N nombre entier positif donné).

Nous avons **trois types** de structures itératives (boucles) :

1 *La boucle*

Pour

Le nombre de répétitions est connu à l'avance.

2 *La boucle*

Tantque

On répète tant que la condition est vraie (testée avant).

3 *La boucle*

Répéter

On répète jusqu'à ce que la condition soit vraie (testée après).

Boucle Pour (For)

La structure de contrôle répétitive **pour** (**for** en langage C) utilise un **indice entier** qui varie (avec un **incrément = 1**) d'une valeur initiale jusqu'à une valeur finale. À la fin de chaque itération, l'indice est incrémenté de 1 d'une manière **automatique (implicite)**.

La syntaxe de la boucle **pour** est comme suit :

Algorithme

```
pour <cpt> ← <vi> à <vf> faire
    <instruction(s)> ;
finPour;
```

Langage C

```
for (<cpt>=<vi>; <cpt><=<vf>; <cpt>++)
{
    <instruction(s)> ;
}
```

`<cpt>` : compteur (variable entière) `<vi>` : valeur initiale `<vf>` : valeur finale

Boucle Pour : fonctionnement

Le bloc d'instructions

La boucle **pour** contient un **bloc d'instructions** (les instructions à répéter).

Si le bloc contient **une seule instruction**, les accolades { et } sont **facultatifs**.

Nombre de répétitions

```
(<vf> - <vi> + 1)
```

si la valeur finale est **supérieure ou égale** à la valeur initiale.

Le bloc sera exécuté pour :

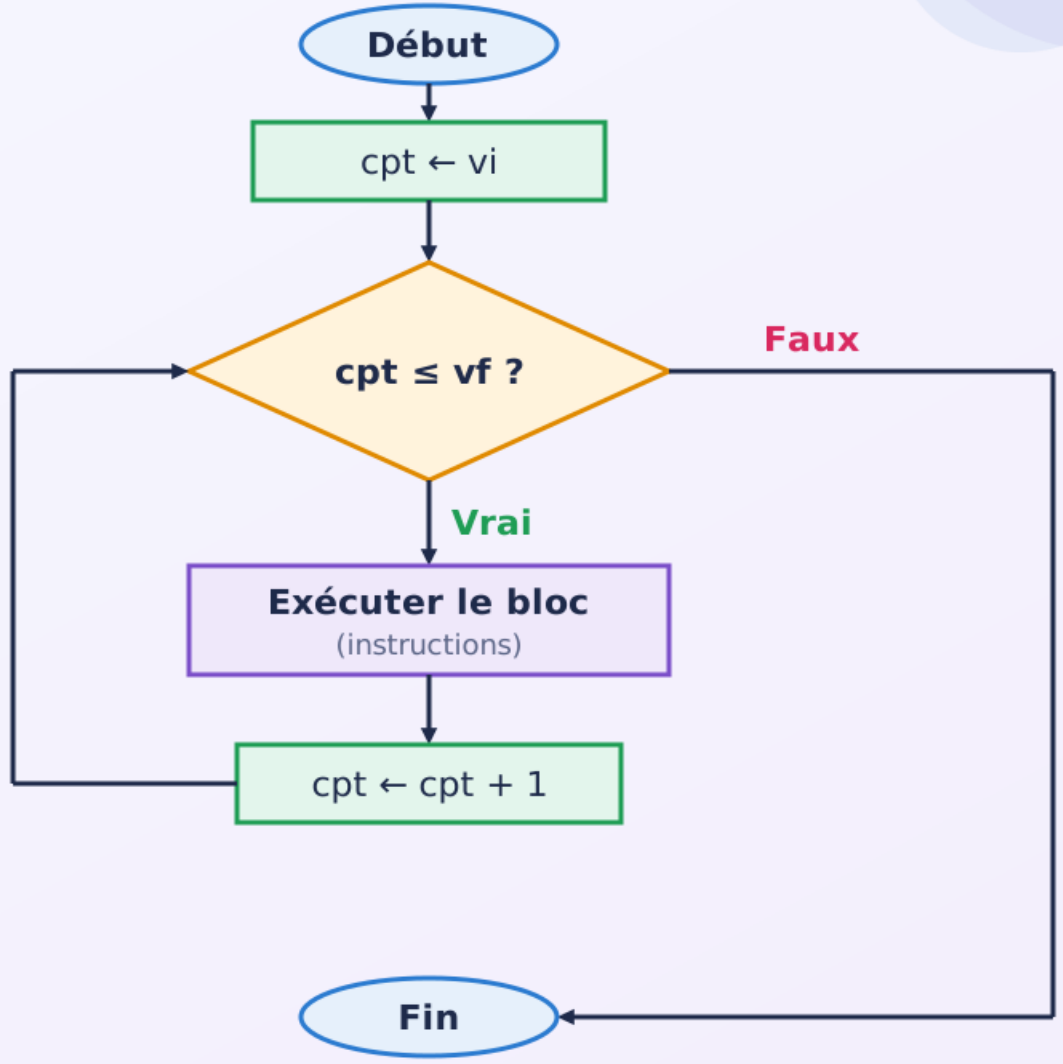
`<cpt> = <vi>`	→	`<cpt> = <vi>+1`	→	`<cpt> = <vi>+2`	→	…	→	`<cpt> = <vf>`
1re itération		2e itération		3e itération				dernière itération

À chaque itération, l'indice est **incrémenté de 1** automatiquement jusqu'à atteindre **<vf>**.

Organigramme de la boucle Pour

Principe

1. Initialisation du compteur :
   `cpt ← vi`
2. Test de la condition :
   `cpt ≤ vf ?`
3. Vrai : exécuter le bloc, puis cpt ← cpt + 1,
   puis retour au test
4. Faux : sortie de la boucle
   et passage à Fin



Somme des N premiers entiers

Énoncé

Boucle **Pour** : calculer la somme suivante

Calculer la **somme** des **N** premiers entiers :

$$S = 1 + 2 + 3 + \cdots + N$$

L'entier **N** est saisi par l'utilisateur ; le résultat **S** est affiché à la fin.

1 Algorithme

Écrire l'algorithme en utilisant une boucle `Pour` avec les variables `N` `i` `S`.

2 Langage C

Traduire l'algorithme en un **programme en langage C** avec `scanf` `for` `printf`.

3 Organigramme

Dessiner l'**organigramme** : début, saisie de N, boucle `i ≤ N ?`, affichage, fin.

Somme des N premiers entiers

Boucle pour : calculer S = 1 + 2 + 3 + ... + N

Algorithme

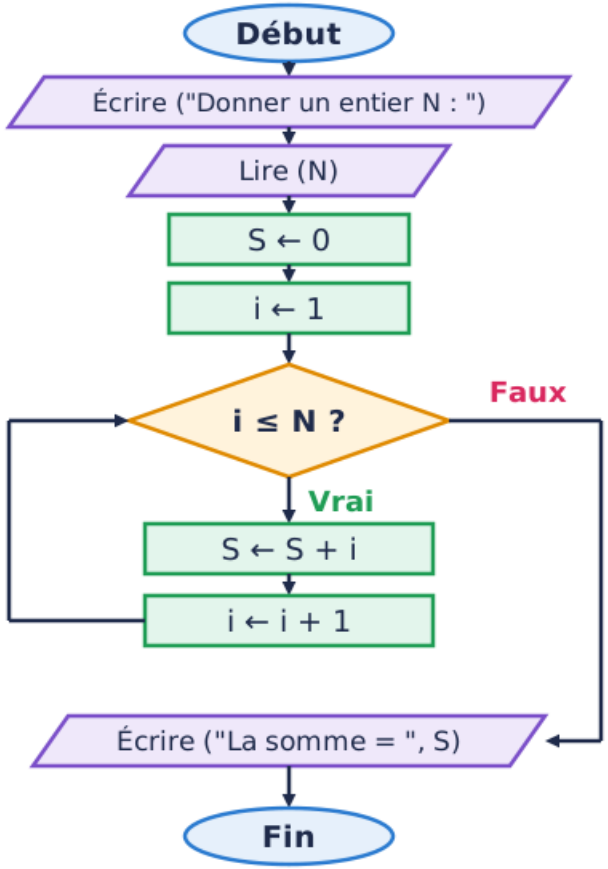
```
Algorithme Somme ;
Variables
    N, i, S : entier ;
Début
    Écrire ("Donner un entier N : ") ;
    Lire (N) ;
    S ← 0 ;
    pour i ← 1 à N faire
        S ← S + i ;
    finPour;
    Écrire ("La somme = ", S) ;
Fin.
```

Langage C

```c
#include <stdio.h>

int main()
{
   int N, i, S;
   printf("Donner un entier N : ");
   scanf("%d", &N);
   S = 0;
   for (i=1; i<=N; i++)
      S = S + i;
   printf("La somme = %d", S);
   return 0;
}
```

Organigramme



Exercice 02

Algorithme → Langage C

Soit l'algorithme suivant : **Exo2**

A Algorithme Exo2

```
Algorithme Exo2 ;
Variables
    X, P, S : réel ;
    i, N : entier ;
Début
    // Entrées
    Ecrire ("Donner les valeurs de N et X : ") ;
    Lire (N, X) ;
    // Traitement
    S ← 0 ;
    P ← X ;
    Pour i ← 1 à N faire
        S ← S + P / i ;
        P ← P * X ;
    Fin-Pour
    // Sortie
    Ecrire ("S= ", S) ;
Fin.
```

1. Traduire l'algorithme en un **programme en langage C**.

2. Dérouler le programme pour `N=3` et `X=3`.

3. Déduire l'**expression générale** du résultat `S` en fonction de `X` et `N` ?

4. Modifier l'algorithme pour calculer la somme **S2** :

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

Corrigé de l'exercice 02

Algorithme → Langage C

1) Traduire l'algorithme en Programme C

A Algorithme

```
Algorithme Exo2 ;
Variables
    X, P, S : réel ;
    i, N : entier ;
Début

    // Entrées
    Ecrire ("Donner les valeurs de N et X : ") ;
    Lire (N, X) ;

    // Traitement
    S ← 0 ;
    P ← X ;
    Pour i ← 1 à N faire
        S ← S + P / i ;
        P ← P * X ;
    Fin-Pour

    // Sortie
    Ecrire ("S = ", S) ;
Fin.
```

C Programme C

```
#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);
    return 0;
}
```

Corrigé de l'exercice 02

Algorithme → Langage C

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

Instructions	Variables					Affichage
	N	X	i	P	S	
`Ecrire ("Donner les valeurs de N et X : ") ;`						Donner les valeurs de N et X :
`Lire (N, X) ;`	3	3	/	/	/	/
`S ← 0 ;`	3	3	/	/	0	/
`P ← X ;` `P ← 3`	3	3	/	3	0	/

`Pour i ← 1`
`S ← S + P/i ;`
`S ← 0 + 3/1 ;` `S ← 3`
`P ← P*X ;`
`P ← 3*3 ;` `P ← 9`
	3	3	1	9	3	

`Pour i ← 2`
`S ← S + P/i ;`
`S ← 0 + 3/1 + 9/2 ;` `S ← 7.5`
`P ← P*X ;`
`P ← 3*3*3 ;` `P ← 27`
	3	3	2	27	7.5	

`Pour i ← 3`
`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`
	3	3	3	81	16.5	

`Ecrire ('S = ', S) ;`	3	3	3	81	16.5	S = 16.500

Corrigé de l'exercice 02

Algorithme → Langage C

3) Déduire l'expression générale de S

Selon le déroulement ci-dessus, nous avons :

i = 1 $S = 3$

i = 2 $S = 3 + \frac{9}{2} = 7.5$

i = 3 $S = 3 + \frac{9}{2} + \frac{27}{3} = 16.5$

i = N $S = X + \frac{X^2}{2} + \frac{X^3}{3} + ... + \frac{X^N}{N}$

On peut **généraliser** par la formule suivante :

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

Corrigé de l'exercice 02

Algorithme → Langage C

4) Modifier l'algorithme pour calculer la somme **S2** :

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

A Algorithme

```
Algorithme Exo2 ;
Variables
    X, P, S : réel ;
    i, N : entier ;
Début

    // Entrées
    Ecrire ("Donner les valeurs de N et X : ") ;
    Lire (N, X) ;

    // Traitement
    S ← 0 ;
    P ← X ;
    Pour i ← 1 à N faire
        S ← S + P / i ;
        P ← P * X * X ;
    Fin-Pour

    // Sortie
    Ecrire ("S =", S) ;
Fin.
```

C Programme C

```
#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);
    return 0;
}
```

Exercice 03

Énoncé → Algorithme → Langage C

Écrire un **algorithme / programme C** pour calculer chacune des sommes / produits suivants :

1 Calculer la **somme**

$$S = 1^2 + 3^2 + 5^2 + \cdots + (2N+1)^2$$

2 Calculer le **produit**

$$P = 1 \times 2 \times 3 \times \ldots \times N$$

3 Calculer la **somme**

$$S = x + \frac{x^3}{2} + \frac{x^5}{4!} + \frac{x^7}{6!} + \cdots + (N\text{ème } terme)$$

4 Calculer la **somme**

$$S = x - x^2 + x^3 - \cdots \mp x^N$$

Corrigé de l'exercice 03

Algorithme → Langage C

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

A Algorithme

```
Algorithme exo3_1 ;
Variables
    S, N, i : entier ;
Début

    // Entrées
    Ecrire ("Donner la valeur de N : ") ;
    Lire (N) ;

    // Traitement
    S ← 0 ;
    Pour i ← 0 à N faire
        S ← S + (2*i+1)² ;
    Fin-Pour

    // Sortie
    Ecrire ("S =", S) ;
Fin.
```

C Programme C

```
#include<stdio.h>
#include<math.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);
        // S=S+(2*i+1)*(2*i+1);

    // Sortie
    printf("S=%d",S);
    return 0;
}
```

Corrigé de l'exercice 03

Algorithme → Langage C

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

A Algorithme

```
Algorithme exo3_2 ;
Variables
    N, i, P : entier ;
Début

    // Entrées
    Ecrire ("Donner la valeur de N : ") ;
    Lire (N) ;

    // Traitement
    P ← 1 ;
    Pour i ← 1 à N faire
        P ← P * i ;
    Fin-Pour

    // Sortie
    Ecrire ("P =", P) ;
Fin.
```

C Programme C

```
#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);
    return 0;
}
```

Corrigé de l'exercice 03

Algorithme → Langage C

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

A Algorithme

```
Algorithme exo3_3 ;
Variables
    N, i, F : entier ;
    X, P, S : réel ;
Début
    // Entrées
    Ecrire ("Donner la valeur de N : ") ;
    Lire (N) ;
    Ecrire ("Donner la valeur de X : ") ;
    Lire (X) ;

    // Traitement
    S ← 0 ;
    P ← X ;
    F ← 1 ;
    Pour i ← 0 à (N-1) faire
        S ← S + P/F ;
        P ← P*X*X ;
        F ← F*(2*i+1)*(2*i+2) ;
    Fin-Pour

    // Sortie
    Ecrire ("S = ", S) ;
Fin.
```

C Programme C

```
#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);
    return 0;
}
```

Corrigé de l'exercice 03

Algorithme → Langage C

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

A Algorithme

```
Algorithme exo3_4 ;
Variables
    N, i : entier ;
    X, P, S : réel ;
Début
    // Entrées
    Ecrire ("Donner la valeur de N : ") ;
    Lire (N) ;
    Ecrire ("Donner la valeur de X : ") ;
    Lire (X) ;

    // Traitement
    S ← 0 ;
    P ← -1 ;
    Pour i ← 1 à N faire
        P ← -P*X ;
        S ← S+P ;
    Fin-Pour

    // Sortie
    Ecrire ("S= ", S) ;
Fin.
```

C Programme C

```
#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);
    return 0;
}
```


Merci