Université A. Mira de Bejaia
Faculté de Technologie
Département de Technologie

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Année universitaire 2025/2026
1ère Année Ingénieur

TP Structure des ordinateurs et applications

Série de TP N°2 –*Les instructions de lecture, écriture et affectation en langage C*

But du TP : Le but de ce travail pratique est d'amener les étudiants à acquérir une compréhension approfondie des **fondamentaux de la programmation en langage C**, à travers la **traduction d'algorithmes simples** en programmes exécutables.

Exercice N°01 :(Algorithme): Soit l'algorithme suivant :

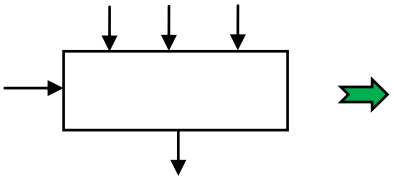
```
Algorithme Exo1 ;
Constantes
   G = 6.674E-11 ; // Constante de gravitation universelle
Variables
   m1, m2, d, F : Réel ;
Début
   // Entrées
   Écrire("Donner la masse m1 (en kg) :") ;
   Lire(m1) ;
   Écrire("Donner la masse m2 (en kg) :") ;
   Lire(m2) ;
   Écrire("Donner la distance d entre les deux objets (en m) :") ;
   Lire(d) ;
   // Traitement
   F ← G * (m1 * m2) / (d * d) ;
   // Sortie
   Écrire("La force gravitationnelle F = ", F, " Newtons") ;
Fin.
```

Questions :

1. Dérouler l'algorithme pour :
   - **m1 = 100 kg** (ex. une personne)
   - **m2 = 10 kg** (ex. un objet)
   - **d = 2 m** (distance entre la personne et l'objet)

Instructions	Variables					Affichage
	m1	m2	d	G	F	
Instruction 1						
Instruction 2						
⋮						
Instruction N						

2. Compléter le schéma suivant :



Entrée :

Traitements :

Sorties :

Exercice N°02 :(Algorithme ➔ Programme en langage C)

Soit l'algorithme suivant:

```
Algorithme Exo2 ;
Constantes
   Pi = 3.1416 ;
Variables
   r, A, P : Réel ;
Début
   Écrire("Donner le…… r du cercle :") ;
   Lire(r) ;
   // Traitement
   A ← Pi * r * r ;
   P ← 2 * Pi * r ;
   // Sorties
   Écrire("……..du cercle est : ", A) ;
   Écrire("……du cercle est : ", P) ;
Fin.
```

Questions:

1. Compléter les instructions d'écriture dans l'algorithme
2. Traduis cet algorithme en un programme en langage C, puis compile et exécute le code. Enfin, fais le déroulement du programme pour r = 5.
3. Que fait ce programme?
4. Quelle sont les nouvelles valeurs de A, P pour r=7 avec approximé à deux décimales?

Exercice N°03 :(Programme en langage C)

Écris un programme en C pour chacun des deux problèmes suivants:

1. Permuter les valeurs de deux variables, ***X*** et ***Y***.
2. Permuter les valeurs de trois variables, ***X***, ***Y*** et ***Z***, de telle sorte que la valeur de ***X*** soit transférée à ***Y***, celle de ***Y*** à ***Z***, et celle de ***Z*** à ***X***.

Exercice N°04 :(Programme en langage C)

Proposer un seul programme en langage C qui permet d'effectuer les opérations suivantes :

1. Le quotient et le reste de la division euclidienne de ***a***=5 par ***b***=2.
2. La somme de ***C***=3 et ***D=4***, ainsi que le produit de ***C*** et ***E***=5.
3. La valeur absolue et le carré d'un nombre réel=2.2.

Exercice N°05 :(Programme en langage C)

Ecris un programme en C qui permet de lire les notes de trois matières (***N1***, ***N2*** et ***N3***), puis calculer et afficher leur moyenne ***M***. Modifiez le programme pour prendre en compte des coefficients (***C1***, ***C2*** et ***C3***) attribués aux trois matières.

TP Structure des ordinateurs et applications

Série de TP N°2 –*Exercices supplémentaires*

Exercice Sup-01 :

1. Ecrire un algorithme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.
2. Écrire un algorithme qui demande à l'utilisateur la valeur de deux nombres A et B puis qui calcule et affiche la somme de leur carré.

Exercice Sup-02 :

Écrire un algorithme qui demande à l'utilisateur la largeur et la longueur d'un rectangle (en mètre) puis qui calcule et affiche le périmètre et l'aire (surface) de ce rectangle.

Exercice Sup-03 :

Exécuter les séquences d'instructions suivantes manuellement et donner les valeurs finales des variables A, B, C et celles de X, Y, Z.

a) A← 5; B← 3; C← B+A; A← 2; B← B+4; C← B-2

b) X← -5; Y← 2*X; X← X+1; Y←sqr(-X-Y); Z ←sqr(-X+Y); X← -(X+3*Y)+2

Écrire les algorithmes correspondants, puis les programmes en langage C correspondants, et fin, procéder à leur exécution.

Exercice Sup-04 :

Écrire un algorithme permettant d'effectuer une permutation circulaire de trois nombres entiers a, b et c.

Exemple :
Si a=10, b=20 et c=30, après permutation, on obtient : a=30, b=10 et c=20.

Exercice Sup-05 :

Ecrire un algorithme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant.

Exercice Sup-06 :

Écrire un algorithme, puis le traduire en langage C, qui permet de calculer le volume d'un cylindre et d'afficher le résultat avec deux puis quatre chiffres après la virgule.

Exercice Sup-07 :

Ecrivez un algorithme qui lit le nom puis le prénom d'un étudiant ensuit affiche le message « Vous vous appelez [NOM] [PRENOM] ».

Université A. Mira de Bejaia
Faculté de Technologie
Département de Technologie

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Année universitaire 2025/2026
1ère Année Ingénieur

TP Structure des ordinateurs et applications

Corrigé de la série de TP N°2–*Les instructions de lecture, écriture et affectation en langage C*

Rappel :

Structure d'un algorithme / programme

Un algorithme manipule des données, les données avant de les utiliser il faut les identifier et les déclarer en utilisant les identificateurs. Un algorithme est constitué de trois parties :

– ***Entête :*** dans cette partie on déclare le nom de l'algorithme à travers un identificateur.

– ***Déclarations :*** dans cette partie on déclare toutes les données utilisées par l'algorithme.

– ***Corps :*** représente la séquence d'actions (instructions)

Pour écrire un algorithme, il faut suivre la structure suivante :

```
Algorithme <id_algorithme>
     <Déclarations>
Début
     <Corps : Instructions>
Fin.
```

```
#include <stdio.h>
int main()
{
     <déclarations>;
     <Instructions>;
     return 0;
}
```

Exercice N°01 :(Algorithme ➔ Programme C)

1) Déroulement de l'algorithme pour **:**

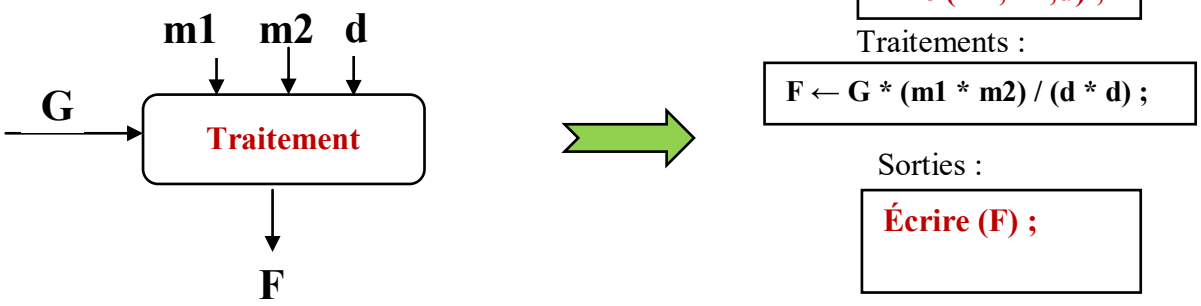
- **m1 = 100 kg** (ex. une personne)
- **m2 = 10 kg** (ex. un objet)
- **d = 2 m** (distance entre la personne et l'objet)

Algorithme

```
Algorithme Exo1 ;
Constantes
   G = 6.674E-11 ; // Constante de gravitation universelle
Variables
   m1, m2, d, F : Réel ;
Début
   // Entrées
   Écrire("Donner la masse m1 (en kg) :") ;
   Lire(m1) ;
   Écrire("Donner la masse m2 (en kg) :") ;
   Lire(m2) ;
   Écrire("Donner la distance d entre les deux objets (en m) :") ;
   Lire(d) ;
   // Traitement
   F ← G * (m1 * m2) / (d * d) ;
   // Sortie
   Écrire("La force gravitationnelle F = ", F, " Newtons") ;
Fin.
```

Instructions	Variables					Affichage
	m1	m2	d	G	F	
				6.674E-11		
Écrire("Donner la masse m1 (en kg) :") ;	/	/	/	//	/	Donner la masse m1 (en kg) :
Lire(m1) ;	100	/	/	//	/	
Écrire("Donner la masse m2 (en kg) :") ;	100	/	/	//	/	Donner la masse m2 (en kg) :
Lire(m2) ;	100	10	/	//	/	
Écrire("Donner la distance d entre les deux objets (en m) :") ;	100	10	/	//	/	Donner la distance d entre les deux objets (en m) :"
Lire(d) ;	100	10	2	//	/	/
F ← G * (m1 * m2) / (d * d) ; F ← 6.674E-11 * (100 * 10) / (2 * 2) ; F ←1.6685E-08;	100	10	2	//	1.6685E-08	/
Écrire("La force gravitationnelle F = ", F, " Newtons") ;	100	10	2	//	//	La force gravitationnelle F = 1.6685e-08 Newtons

2)Le schéma complet est :



Exercice N°02 :(Algorithme ➔ Programme C)

Algorithme	***Questions:***
Algorithme Exo2 ; ***Constantes*** ***Pi = 3.1416 ;*** ***Variables*** ***r, A, P : Réel ;*** ***Début*** ***Écrire("Donner le rayon r du cercle :") ;*** ***Lire(r) ;*** ***// Traitement*** ***A ← Pi * r * r ;*** ***P ← 2 * Pi * r ;*** ***// Sorties*** ***Écrire("L'aire du cercle est : ", A) ;*** ***Écrire("Le périmètre du cercle est : ", P) ;*** ***Fin.***	1. Compléter les instructions d'écriture dans l'algorithme 2. Traduis cet algorithme en un programme en langage C, puis compile et exécute le code. Enfin, fais le déroulement du programme pour r = 5. 3. Que fait ce programme? 4. Quelle sont les nouvelles valeurs de A, P pour r=7 avec approximé à deux décimales?

1) Compléter les instructions d'écriture dans l'algorithme (voir l'algorithme)

2) Traduis cet algorithme en un programme en langage C, puis compile et exécute le code. Enfin, fais le déroulement du programme pour r = 5.

Algorithme	Programme C
Algorithme Exo2 ; *Constantes* *Pi = 3.1416 ;* *Variables* *r, A, P : Réel ;* *Début* *// Entrèes* *Écrire("Donner le rayon r du cercle :") ;* *Lire(r) ;* *// Traitement* *A ← Pi * r * r ;* *P ← 2 * Pi * r ;* *// Sorties* *Écrire("L'aire du cercle est : ", A) ;* *Écrire("Le périmètre du cercle est : ", P) ;* *Fin.*	*#include <stdio.h>* *int main() {* *const float Pi = 3.1416; // constante de type réelle* *float r, A, P;* *// Entrèes* *// Demander à l'utilisateur le rayon du cercle* *printf("Donner le rayon r du cercle : ");* *scanf("%f", &r);* *// Traitement* *// Calcul de l'aire et du périmètre* *A = Pi * r * r; // A = Pi * pow(r, 2);* *P = 2 * Pi * r;* *// Sorties* *// Affichage des résultats* *printf("L'aire du cercle est : %f\n", A);* *printf("Le périmètre du cercle est : %f\n", P);* *return 0;* *}*

```c
#include <stdio.h>
int main() {
    const float Pi = 3.1416;  // constante de type réelle
    float r, A, P;
    // Entrèes
    // Demander à l'utilisateur le rayon du cercle
    printf("Donner le rayon r du cercle : ");
    scanf("%f", &r);
    // Traitement
    // Calcul de l'aire et du périmètre
    A = Pi * r * r; // A = Pi * pow(r, 2);
    P = 2 * Pi * r;
    // Sorties
    // Affichage des résultats
    printf("L'aire du cercle est : %f\n", A);
    printf("Le périmètre du cercle est : %f\n", P);
    return 0;
}
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\exoo55...
Donner le rayon r du cercle : 5
LÆaire du cercle est : 78.540001
Le pÚrimÞtre du cercle est : 31.415998

Process returned 0 (0x0)   execution time : 2.000 s
Press any key to continue.
```

Déroulement du programme C pour r=5

Instructions	Variables			Affichage
	r	A	P	
printf("Donner le rayon r du cercle : ");	/	/	/	*Donner le rayon r du cercle:*
scanf("%f", &r);	5	/	/	/
*A = Pi * r * r;* *A = 78.540001;*	5	78.540001	/	/
*P = 2 * Pi * r;* *P = 31.415998;*	5	78.540001	*31.415998*	/

printf("L'aire du cercle est : %f\n", A);	5	78.540001	31.415998	*L'aire du cercle est :* 78.540001
printf("Le périmètre du cercle est : %f\n", P);	5	78.540001	31.415998	*Le périmètre du cercle est : 31.415998*

3) Que fait ce programme?

Ce programme permet de calculer l'aire et le périmètre d'un cercle à partir du rayon saisi par l'utilisateur.

4) Quelle sont les nouvelles valeurs de A, P pour r=7 avec approximé à deux décimales?

Après avoir compiler et executer le programme une nouvelle fois pour r=7 en modifiant %f par %.2f dans les deux instructions de lecture printf:
Aire : A=153.94 et Périmètre : P=43.98

```c
#include <stdio.h>
int main() {
    const float Pi = 3.1416;  // constante de type réelle
    float r, A, P;
    // Entrèes
    // Demander à l'utilisateur le rayon du cercle
    printf("Donner le rayon r du cercle : ");
    scanf("%f", &r);
    // Traitement
    // Calcul de l'aire et du périmètre
    A = Pi * r * r; // A = Pi * pow(r, 2);
    P = 2 * Pi * r;
    // Sorties
    // Affichage des résultats
    printf("L'aire du cercle est : %.2f\n", A);
    printf("Le périmètre du cercle est : %.2f\n", P);
    return 0;
}
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\exoo55\bin\Debug\exoo55....
Donner le rayon r du cercle : 7
LÆaire du cercle est : 153.94
Le pÚrimÞtre du cercle est : 43.98

Process returned 0 (0x0)   execution time : 2.750 s
Press any key to continue.
```

Exercice N°03 :(Programme en langage C)

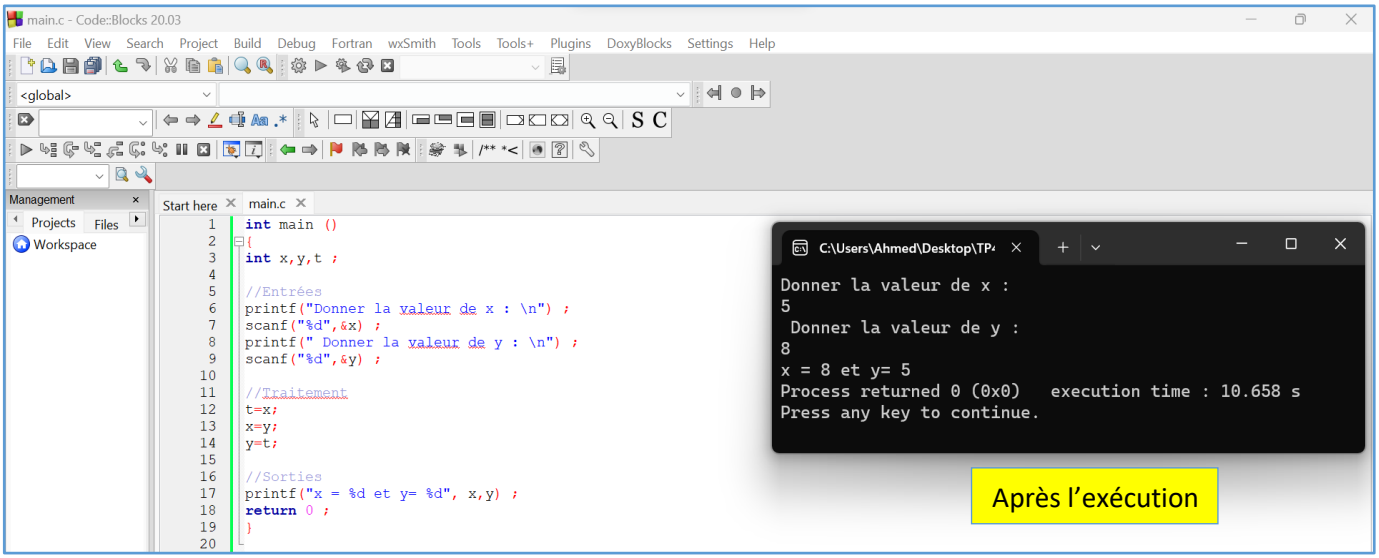
Écris un programme en C pour chacun des deux problèmes suivants:

1. Permuter les valeurs de deux variables, ***X*** et ***Y***.
2. Permuter les valeurs de trois variables, ***X***, ***Y*** et ***Z***, de telle sorte que la valeur de ***X*** soit transférée à ***Y***, celle de ***Y*** à ***Z***, et celle de ***Z*** à ***X***.

1) Permutation entre les deux variables X et Y :

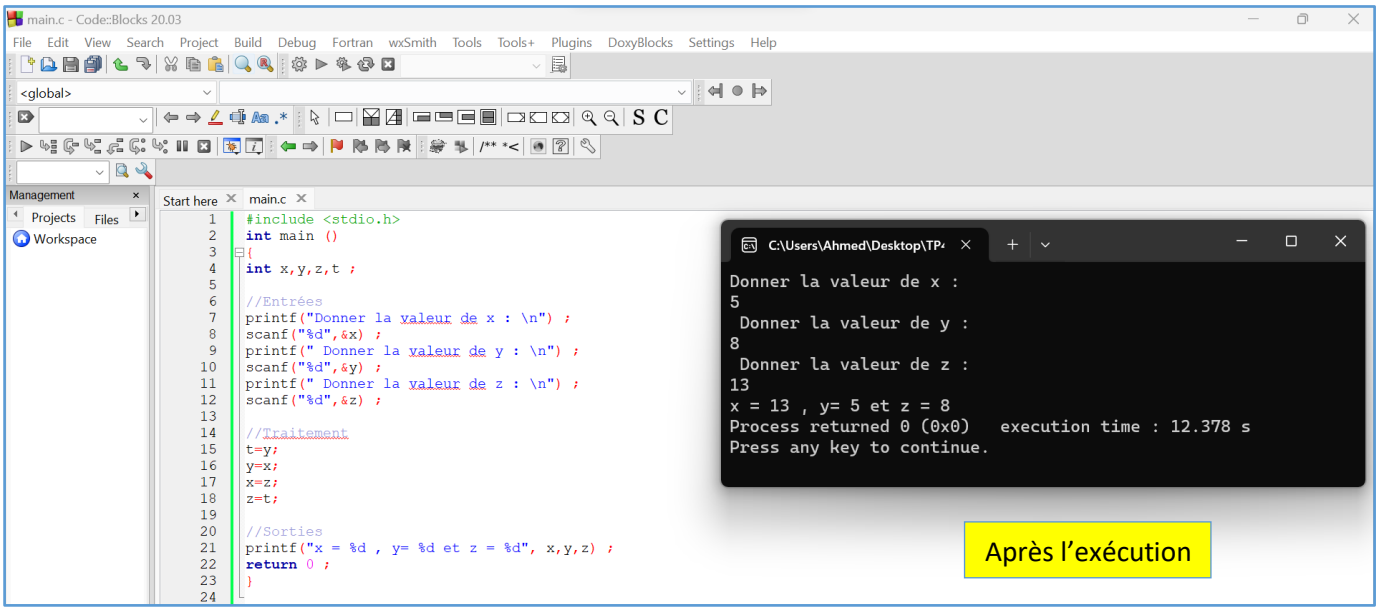
Algorithme	**Programme C**
Algorithme Exo3_a; **Variables** x, y, t : **entier**; **Début** //Entrées **Ecrire**("Donner la valeur de x : ") ; **Lire**(x) ;	`#include <stdio.h>` `int main( )` `{` `  int x,y,t;` `  //Entrées` `  printf("Donner la valeur de x : \n") ;` `  scanf("%d",&x) ;`

Ecrire("Donner la valeur de y : ") ; **Lire**(y) ; //Traitement t← x; x← y; y← t ; //Sorties Écrire("x=", x, "y=", y) ; **Fin.**	**printf**(" Donner la valeur de y : \n") ; **scanf**("%d",&y) ; //Traitement t=x; x=y; y=t; //Sorties **printf**("x = %d et y= %d", x,y) ; **return 0;** }



2) Permutation entre les trois variables X, Y et Z de telle sorte que la valeur de X soit dans Y, celle de Y dans Z et la valeur de Z dans X :

Algorithme	Programme C
Algorithme Exo3_b; **Variables** x, y, z, t : **entier**; **Début** //Entrées **Ecrire**("Donner la valeur de x : ") ; **Lire**(x) ; **Ecrire**("Donner la valeur de y :") ; **Lire**(y) ; **Ecrire**("Donner la valeur de z : ") ; **Lire**(z) ; //Traitement t ← y; y ← x; x ← z ; z ← t ; //Sorties **Écrire**("x=", x, "y=", y, "z=",z); **Fin.**	#include <stdio.h> int **main( )** { **int** x,y,z,t; //Entrées **printf**("Donner la valeur de x : \n") ; **scanf**("%d",&x) ; **printf**(" Donner la valeur de y : \n") ; **scanf**("%d",&y) ; **printf**(" Donner la valeur de z : \n") ; **scanf**("%d",&z) ; //Traitement t=y; y=x; x=z; z=t; //Sorties **printf**("x = %d , y= %d et z = %d", x,y,z) ; **return 0;** }



Exercice N°04 :(Programme en langage C)

Proposer un seul programme en langage C qui permet d'effectuer les opérations suivantes :

1. Le quotient et le reste de la division euclidienne de ***a***=5 par ***b***=2.
2. La somme de ***C***=3 et ***D=4***, ainsi que le produit de ***C*** et ***E***=5.
3. La valeur absolue et le carré d'un nombre réel=2.2.

1) PROGRAMME EN C :

Algorithme	**Programme C**
Algorithme Exo4; **Variables** a, b, Q, R : **entier**; C, D, E, somme, produit: **entier**; x, valeur_absolue, carre : **de réels;** **Début** //Entrées **Ecrire("Donner la valeur de a et b : ") ;** **Lire** (a,b) ; **Ecrire ("Donner trois entiers C, D et E : ");** **Lire** (C, D, E); **Ecrire ("Donner un nombre réel x : ");** **Lire (x);** //Traitement **Q ← a div b; //div : division entière** **R ← a mod b; //mod : reste de division** **somme ← C + D;** **produit ← C * E;** **valeur_absolue ← |x|;** **carre ← x^2; *//ou bien x.x;*** //Sorties **Écrire**("Q=", Q, "et R=", R) ; **Écrire ("*Somme de C et D =*",somme,** **"*Produit de C et E =* ",produit);** **Écrire ("*Valeur absolue de x=*",** **valeur_absolue, "*Carré de x =* ",carre);** **Fin.**	**#include <stdio.h>** **#include <math.h>** **int main( )** **{ int** a,b,R,Q; **int** C, D, E, somme, produit; **float** x, valeur_absolue, carre; //Entrées **printf("Donner la valeur de a et b: \n") ;** **scanf**("%d%d",&a,&b) ; **printf("Donner trois entiers C, D et E : ");** **scanf("%d%d%d", &C, &D, &E);** **printf("Donner un nombre reel x : ");** **scanf("%f", &x);** //Traitement **Q=a/b;** **R=a%b;** // Le modulo (%) = le reste d'une division. **somme = C + D;** **produit = C * E;** **valeur_absolue = fabs(x);** *// Valeur absolue pour réel* **carre = x * x;** ***// carre = pow(x, 2);*** //Sorties *// Affichage des résultats* **printf("\n Resultats : \n");** **printf("*Q = %d, R = %d\n*", Q, R);** **printf("*Somme de C et D = %d, Produit de C et E = %d\n*",** **somme, produit);** **printf("*Valeur absolue de x = %.2f, Carre de x = %.2f\n*",** **valeur_absolue, carre);** **return 0;** **}**

```c
#include <stdio.h>
#include <math.h>
int main( )
{
int a,b,R,Q;
int C, D, E, somme, produit;
float x, valeur_absolue, carre;
//Entrées
printf("Donner la valeur de a et b: \n") ;
scanf("%d%d",&a,&b) ;
printf("Donner trois entiers C, D et E : ");
scanf("%d%d%d", &C, &D, &E);
printf("Donner un nombre reel x : ");
scanf("%f", &x);
//Traitement
Q=a/b;
R=a%b; // Le modulo (%) = le reste d'une division.
somme = C + D;
produit = C * E;
valeur_absolue = fabs(x); // Valeur absolue pour réel
carre = x * x;  // carre = pow(x, 2);
//Sorties
// Affichage des résultats
    printf("\nResultats :\n");
    printf("Q = %d, R = %d\n", Q, R);
    printf("Somme de C et D = %d, Produit de C et E = %d\n", somme, produit);
    printf("Valeur absolue de x = %.2f, Carre de x = %.2f\n", valeur_absolue, carre);
return 0;
}
```

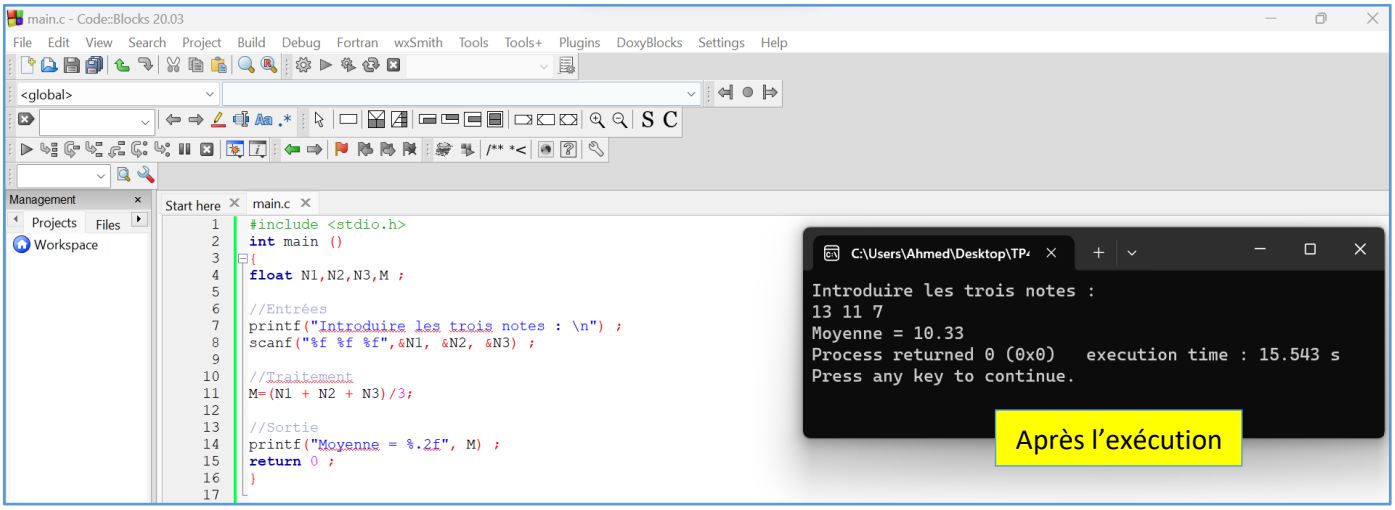
```
C:\Users\Maison\Desktop\lhklklgljgljgj\exoo55\bin\De...
Donner la valeur de a et b:
5 2
Donner trois entiers C, D et E : 3 4 5
Donner un nombre reel x : 2.2

Resultats :
Q = 2, R = 1
Somme de C et D = 7, Produit de C et E = 15
Valeur absolue de x = 2.20, Carre de x = 4.84

Process returned 0 (0x0)   execution time : 11.868 s
Press any key to continue.
```

Exercice N°05 :(Programme en langage C)

Ecris un programme en C qui permet de lire les notes de trois matières (***N1***, ***N2*** et ***N3***), puis calculer et afficher leur moyenne ***M***. Modifiez le programme pour prendre en compte des coefficients (***C1***, ***C2*** et ***C3***) attribués aux trois matières.

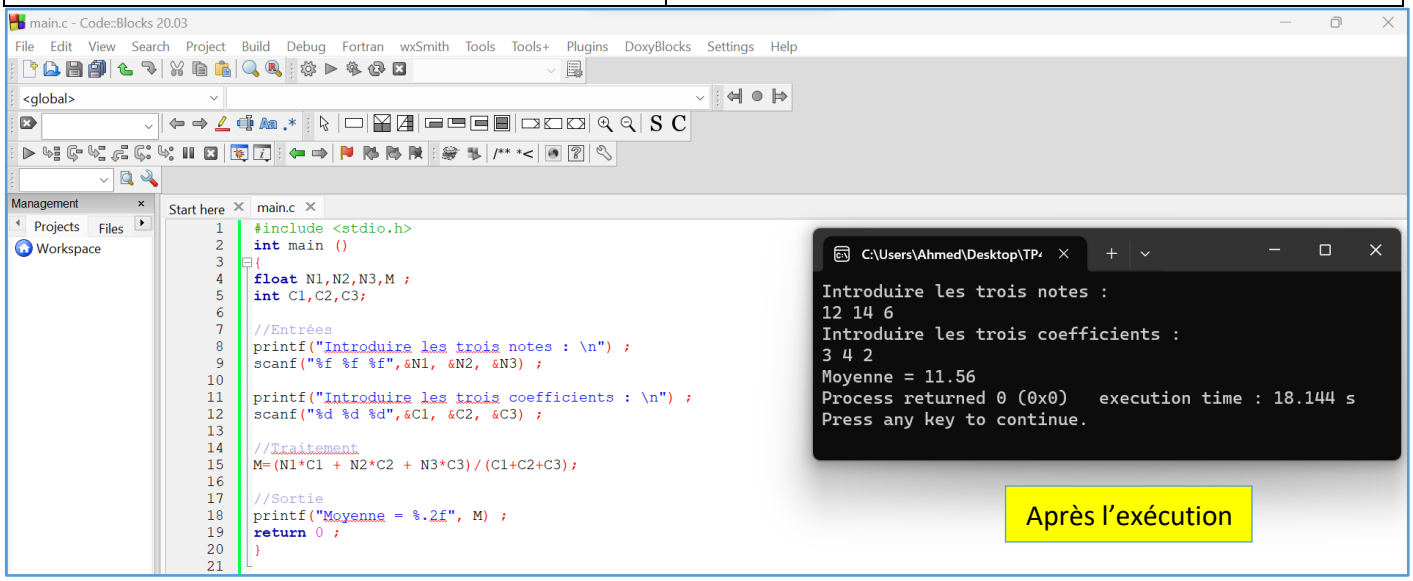
5.a) Programme en C qui permet de lire les notes de trois matières (N1, N2 et N3), puis calculer et afficher leur moyenne M

Algorithme	**Programme C**
Algorithme Exo5_a; **Variables** N1, N2, N3, M :**réel** ; **Début** //Entrées **Écrire** ("Introduire les trois notes : ") ; **Lire**(N1,N2, N3); //Traitement M ← (N1 + N2 + N3)/3; //Sorties **Écrire**("Moyenne =",M:0:2); **Fin.**	**#include <stdio.h>** **int main( )** { **float** N1,N2,N3,M; //Entrées **printf**("Introduire les trois notes :\n") ; **scanf**("%f %f %f",&N1,&N2, &N3) ; //Traitement M=(N1 + N2 + N3)/3; //Sortie **printf**("Moyenne = %.2f", M) ; **return 0;** }



5.b) Modification de l'algorithme dans le cas où des coefficients (C1, C2 et C3) sont attribués aux trois matières :

Algorithme	Programme C
Algorithme Exo5_b; **Variables** N1, N2, N3, M :**réel** ; C1, C2, C3 : **entier** ; **Début** //Entrées **Écrire** ("Introduire les trois notes : ") ; **Lire**(N1,N2, N3); **Écrire** ("Introduire les trois coefficients : ") ; **Lire**(C1,C2, C3); //Traitement M ←(N1*C1 + N2*C2 + N3*C3)/(C1+C2+C3); //Sorties **Écrire**("Moyenne =",M:0:2); **Fin.**	#include <stdio.h> int main( ) { floatN1,N2,N3,M ; intC1,C2,C3; //Entrées printf("Introduire les trois notes :\n") ; scanf("%f %f %f",&N1, &N2, &N3) ; printf("Introduire les trois coefficients :\n") ; scanf("%d %d %d",&C1, &C2, &C3) ; //Traitement M=(N1*C1 + N2*C2 + N3*C3)/(C1+C2+C3); //Sortie printf("Moyenne = %.2f", M) ; return 0; }



Université A. Mira de Bejaia
Faculté de Technologie
Département de Technologie

Année universitaire 2025/2026
1ère Année Ingénieur
Semestre 1

TP Structure des Ordinateurs et Applications

Série de TP N°3– Les instructions de test : Si…Fin-Si & Si…Sinon…Fin-Si

But de TP :
Le but du TP est de permettre aux étudiants de comprendre et de maîtriser les structures de contrôle conditionnelles en programmation.

Exercice N°01 : (**Objectif :** Se familiariser avec la syntaxe de la structure if ... else).

Soit l'algorithme suivant :

```
Algorithme exo1 ;
Variables
        x , abs: entier ;
Début
 // Entrées
  Écrire ("Donner la valeur de x : ") ;
  Lire(x) ;
 // Traitements
    si (x < 0) alors
       abs = -x ;
    sinon
       abs = x ;
    fin-si
 //sorties
  Écrire (" La valeur de x =  ",abs) ;
Fin.
```

1. Traduire l'algorithme en un programme en langage C.
2. Compiler et exécuter le programme pour les valeurs suivantes :
   - x = 5 ;
   - x = - 8 ;
3. Dérouler l'algorithme pour les cas suivants :
   - x = 5 ;
   - x = -8 ;
4. Déduire ce que fait cet algorithme ?

Exercice N°02 : (**Objectif :** utiliser une structure conditionnelle à plusieurs branches if ... else if ... else).

Écrire un programme en langage C qui permet de calculer les racines d'un polynôme du second degré de la forme :

$$ax^2 + bx + c = 0$$

Rappel : Le discriminant Δ est donné par la formule :$\Delta = b^2 - 4ac$

- Si Δ > 0 → deux racines réelles et distinctes : $x1 = (-b - \sqrt{\Delta})/\ 2a$; $x1 = (-b + \sqrt{\Delta})/\ 2a$;
- Si Δ = 0 → une racine réelle double : $x = -b/\ 2a$;
- Si Δ < 0 → pas de racine réelle (racines complexes).

- Compiler pour les valeurs suivantes : a = 1, b = 5, c =6 ; a = 1, b = 2, c = 1 ; a = 1, b = 2, c = 5 ;

Exercice N°03 :(**Objectif :** appliquer les opérateurs logiques (**ET**) et (**OU**)).

Écrire un programme en langage C qui permet de vérifier si un triangle est **équilatéral**, **isocèle** ou **quelconque**.

Rappel : équilatéral → (a = b **ET** b = c), Isocèle → (a = b **OU** a = c **OU** b = c), Quelconque → tous les autres cas.

- Compiler pour les valeurs suivantes : a = 5, b = 5, c =5 ; a = 4, b = 3, c = 4 ; a = 3, b = 4, c = 5 ;

Exercice N°04 :(**Objectif :** utilisation des conditions dans un exemple concret (TP : Structure de la matière)).

Écrire un programme en langage C qui lit la température de l'eau en degrés Celsius et affiche son état physique.

- **État solide (glace)** → si T < 0 °C ;
- **Fusion** → si T = 0 °C ;
- **État liquide** → si T <100 °C ;
- **Ébullition** → si T = 100 °C ;
- État gazeux (vapeur) → si T > 100 °C ;

Exercice N°05 :(**Objectif :** utilisation des conditions dans un exemple concret (Analyse 1)).

Écrire un programme en langage C qui permet de vérifier si une suite composée de trois valeurs A, B et C est monotone décroissante.

Rappel une suite monotone décroissante (A > B et B > C).

Série de TP N°3 - Exercices supplémentaires et rappels

Rappel : syntaxe des structures conditionnelles

simple	Double	imbriquées
if (Condition) { <bloc_instructions_if> ; }	**if** (Condition) { <bloc_instructions_if> ; } **else** { <bloc_instructions_else> ; }	**if** (Condition) { <bloc_instructions_if> ; } **else if** (Condition) { <bloc_instructions_if> ; } **else** { <bloc_instructions_else> ; }

Rappel :opérateurs de comparaison :
Égal à == , Différent de != , Supérieur à > , Inférieur à< , Supérieur ou égal à>= , Inférieur ou égal à<=.
Opérateurs logiques :
ET logique && , **OU logique||** .

Exercice Sup-01 :

Écrire un programme en langage C qui permet à l'utilisateur de saisir un nombre entier. Le programme devra ensuite déterminer si ce nombre est pair ou impair.

Exercice Sup-02:

Écrire un algorithme/programme en langage C qui permet de lire deux nombres entiers (a, b), calcule et affiche la valeur de c comme suit :

$$c = \begin{cases} 0 \; si \; a = b \\ a.b \; si \; a > b \\ a + b \; si \; a < b \end{cases}$$

Exercice Sup-03:

Écrire un programme en langage C qui permet de déterminer le prix d'un article en fonction de plusieurs critères :

- Si le prix initial de l'article est inférieur à 100 D.A :
- Si la quantité achetée est inférieure à 5, appliquer une remise de 5%.
- Sinon, appliquer une remise de 10%.
- Si le prix initial de l'article est compris entre 100 D.A et 500 D.A :
- Si la quantité achetée est inférieure à 10, appliquer une remise de 15%.
- Sinon, appliquer une remise de 20%.
- Si le prix initial de l'article est supérieur à 500 D.A:
- Appliquer une remise de 25% sur le prix initial.

Exercice Sup-04:

Écrire un programme en langage C qui permet à l'utilisateur de saisir un nombre entier. Le programme devra ensuite déterminer si ce nombre est pair ou impair.

Exercice Sup-05:

Écrire un programme en langage C qui lit deux nombres entiers et affiche le plus grand des deux.

Université A. Mira de Bejaia
Faculté de Technologie
Département de Technologie

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Année universitaire 2025/2026
1ère Année Ingénieur
Semestre 1

TP Structure des Ordinateurs et Applications

Corrigé de la Série de TP N°3– Les instructions de test : Si…Fin-Si & Si…Sinon…Fin-Si

Rappel :

Structures de contrôle conditionnel

Les structures conditionnelles sont utilisées pour contrôler l'exécution d'un bloc d'instructions : elles permettent soit d'exécuter ce bloc, soit de le sauter. Elles servent également à choisir entre l'exécution de deux blocs distincts, en fonction de conditions spécifiques.

Syntaxe des structures conditionnelles :

simple	Double	imbriquées
if (Condition) { <bloc_instructions_if> ; }	**if** (Condition) { <bloc_instructions_if> ; } **else** { <bloc_instructions_else> ; }	**if** (Condition) { <bloc_instructions_if> ; } **else if** (Condition) { <bloc_instructions_if> ; } **else** { <bloc_instructions_else> ; }

Rappel :opérateurs de comparaison :
Égal à == , Différent de != , Supérieur à > , Inférieur à< , Supérieur ou égal à>= , Inférieur ou égal à<=.
Opérateurs logiques : **ET logique && , OU logique|| .**

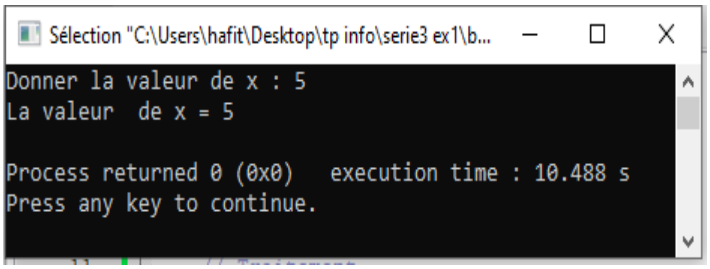
Exercice N°01 ::

1. ***Traduction du programme :***

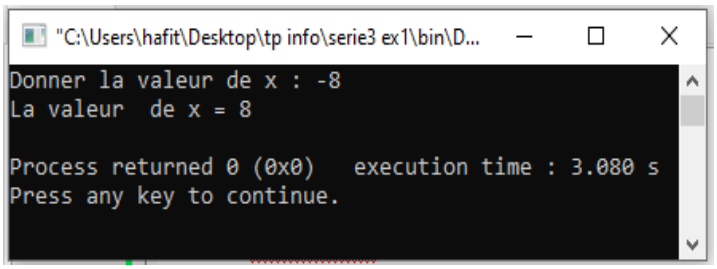
Algorithme	Langages C
Algorithme exo1 ; **Variables** x , abs: entier ; **Début** // Entrées **Écrire**("Donner la valeur de x : ") ; **Lire**(x) ; // Traitements **si** (x < 0) **alors** abs = -x ; **sinon** abs = x ; **fin-si** //sorties **Écrire**(" La valeur de x = ",abs) ; **Fin.**	`#include <stdio.h>` `int main ()` `{` `int x,abs;` `// Entrée` `printf("Donner la valeur de x : ");` `scanf("%d",&x);` `// Traitement` `if(x <0)` `abs=-x;` `else` `abs= x;` `// Sortie` `printf("La valeur de x = %d\n",abs);` `return0;` `}`

2. ***Compiler et exécuter le programme pour :***

➢ ***x = 5 :***



➢ ***x = - 8***



3. ***Dérouler l'algorithme :***

➢ ***Pour x = 5.***

Instructions	***Variables***		***affichages***
	x	***abs***	
Écrire("Donner la valeur de x : ") ;	/	/	Donner la valeur de x :
Lire(x) ;	***5***	/	/
si (x < 0) **alors**(FAUX) abs = x ;	***5***	***5***	/
Écrire(" la valeur de x = ",abs) ;	***5***	***5***	la valeur de x = 5

➢ ***Pour x = -8.***

Instructions	***Variables***		***affichages***
	x	***abs***	
Écrire("Donner la valeur de x : ") ;	/	/	Donner la valeur de x :
Lire(x) ;	***-8***	/	/
si (x < 0) **alors**(VRAI) abs = -x ;	***-8***	***8***	/
Écrire(" la valeurde x = ",abs) ;	***-8***	***8***	la valeur de x = 8

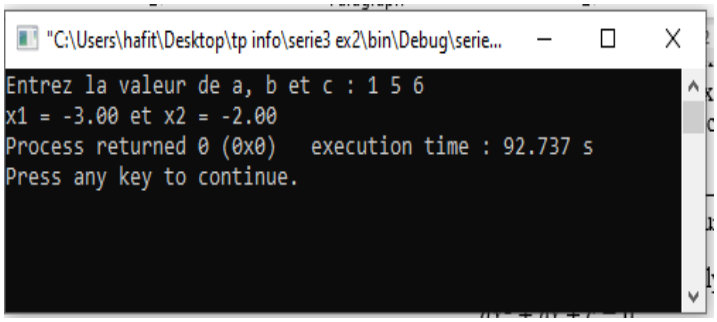
4. ***Déduire ce que fait cet algorithme ?*** **Le programme calcule la valeur absolue d'un nombre x.**

Exercice N°02 :

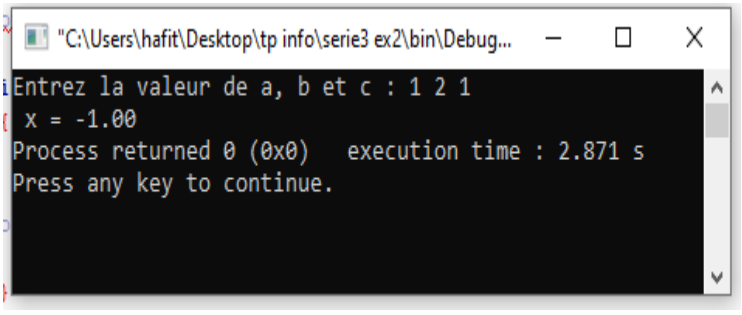
Algorithme	Langage C
Algorithme Exo2 ; **Variables** a, b, c, d, x, x1, x2 : **réel** ; **Début** **Écrire** (" Entrez la valeur de a, b, et c :") ; // Entrées **Lire** (a, b, c) ; // Traitement d ← sqr(b) – 4* a* c ; **Si** (d > 0) **alors** x1 ← (-b - sqrt(d)) / (2*a) ; x2 ← (-b + sqrt(d)) / (2*a) ; // Sorties **Écrire** ("x1=", x1, "x2=", x2) ; **Sinon** **Si** (d=0) alors X ← -b / (2*a); // sortie **Écrire** ('x=', x) ; **Sinon** // sortie Écrire ("Pas de solution réelle") ; **Fin-si ;** **Fin-si ;** **Fin.**	`#include <stdio.h>` `#include <math.h>` `int main(){` `float a, b, c, d, x, x1, x2;` `// Entrées` `printf("Entrez la valeur de a, b et c:");` `scanf("%f%f%f",&a,&b,&c);` `// Traitement` `d =pow(b,2)-4* a * c; //ou(d=b*b -4* a * c;)` `if(d >0)` `{` `x1 =(-b -sqrt(d))/(2* a);` `x2 =(-b +sqrt(d))/(2* a);` `printf("x1 = %.2f x2 = %.2f", x1, x2);` `}` `else if(d ==0)` `{` `x =-b /(2* a);` `printf(" x = %.2f", x);// Sortie` `}` `else` `{` `printf ("Pas de solution reelle");` `}` `return 0;` `}`

➢ ***Compiler pour les valeurs suivantes :***

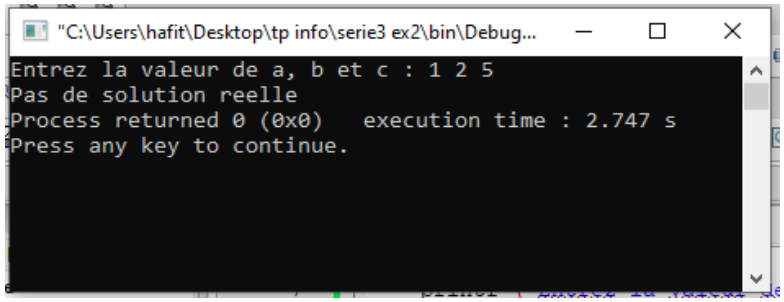
- ***a = 1 , b = 5 , c =6***



- ***a = 1 ,b = 2 , c = 1***



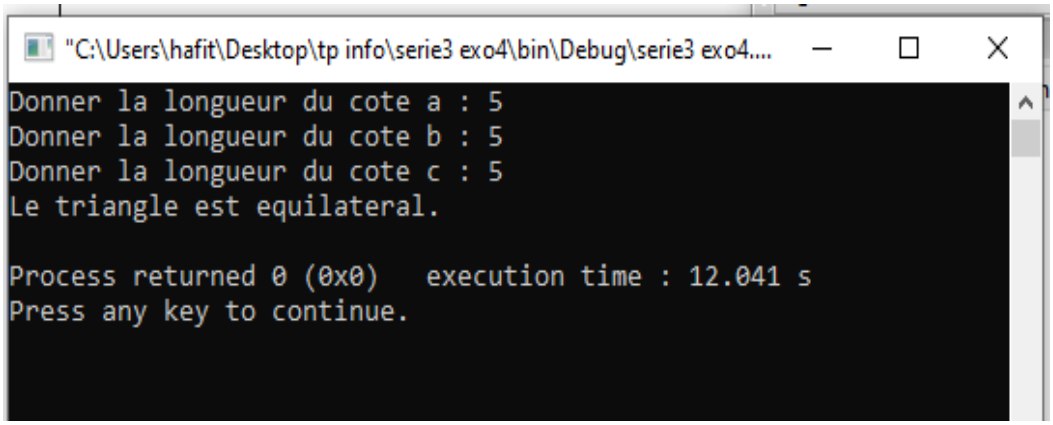
- ***a = 1 , b = 2, c = 5 ;***



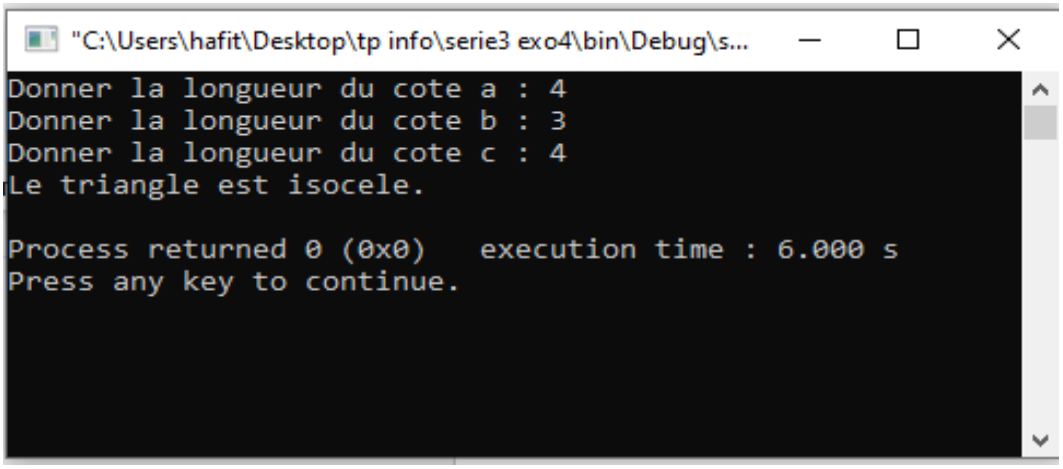
Exercice N°03 :

Programme :

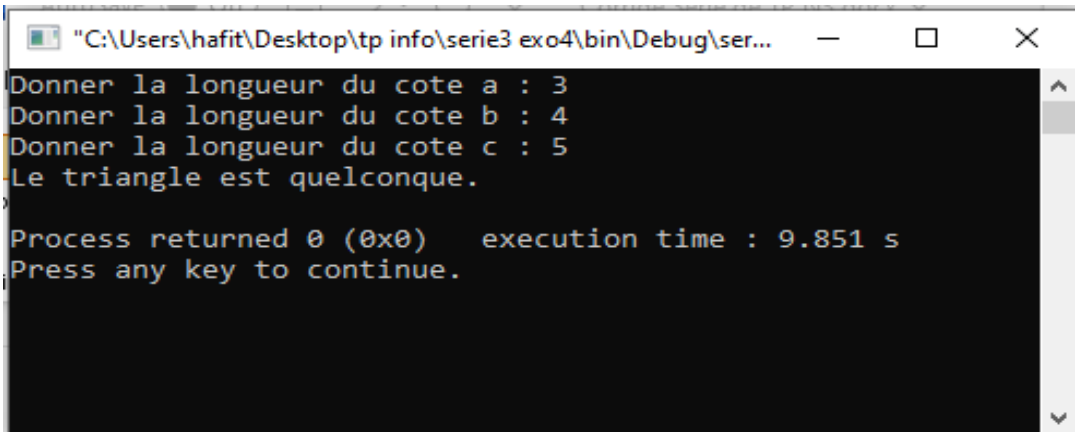
```
#include <stdio.h>

int main ( )
{
    float a, b, c;
   printf("Donner la longueur du cote a : ");
   scanf("%f",&a);
   printf("Donner la longueur du cote b : ");
   scanf("%f",&b);
   printf("Donner la longueur du cote c : ");
   scanf("%f",&c);

  if(a == b && b == c)
     {
       printf("Le triangle est equilateral.\n");
     }
  elseif(a == b || a == c || b == c)
     {
      printf("Le triangle est isocele.\n");
     }
  else
       {
      printf("Le triangle est quelconque.\n");
       }
return0;
}
```



➢ ***Compilation et exécution du programme pour les valeurs suivantes : a = 4 , b = 3, c=4***



➢ ***Compilation et exécution du programme pour les valeurs suivantes : a = 3 , b = 4, c=5***



Exercice N°04 :

Programme :

```c
#include <stdio.h>

intmain()
{
   float T;
   printf("Entrer la temperature en degres Celsius : ");
   scanf("%f",&T);
  if(T <0)
    {
     printf("Etat solide (glace)\n");
    }
  elseif(T ==0)
      {
      printf("Fusion \n");
      }
  elseif(T >0&& T <100)
      {
      printf("Etat liquide (\n");
      }
  elseif(T ==100)
      {
      printf("Ebullition \n");
      }
  else// T > 100
      {
      printf("Etat gazeux (vapeur)\n");
      }
```

Exercice N°05 :

Programme :

```c
#include <stdio.h>

Int main()
{
    float A, B, C;
    printf("Donner la valeur de A : ");
    scanf("%f",&A);
    printf("Donner la valeur de B : ");
    scanf("%f",&B);
    printf("Donner la valeur de C : ");
    scanf("%f",&C);

  if(A > B && B > C)
     {
      printf("La suite est monotone decroissante : A > B > C\n");
     }
  else
     {
      printf("La suite n'est pas monotone decroissante.\n");
      }

  return0;
}
```

TP Structure des Ordinateurs et Applications

Série de TP N°4– *Les instructions itératives : Pour, Tant-que et Répéter*

Exercice N°01 : soit l'algorithme suivant :

Algorithme exercice01 ; Variables N,i : Entier ; R,j : Réel ; Début Lire (N) ; R ←1; j ← 1; Pour i ← 1 à N faire R ← i / j * R ; j← i + 2 ; fin-pour ; écrire ('R= ',R) ; fin.	**Questions :** 1. Traduire l'algorithme donné en programme C, afficher le résultat avec deux chiffres après la virgule. 2. Dérouler l'algorithme pour N=3. 3. Déduire l'expression générale de R. 4. Réécrire le programme C en remplaçant la boucle **pour** par la boucle **tant-que,** ensuite par la boucle **répéter**.

Exercice N°02 :

Ecrire un programme C pour calculer la somme/produit suivants :

$P = 1*2*3*4*5*.....*N$ ← ***Compiler et exécuter le programme pour N=6***

$S = \sum_{i=0}^{N-1} \frac{X^{2i+1}}{(2i+1)!}$ ← ***Compiler et exécuter le programme pour X=2 et N=3***

Exercice N°03 :

Écrire un programme C qui calcule la somme des nombres entiers de 0 à N et le produit des nombres entiers de 1 à N, puis compiler et exécuter le programme pour ***N=15.***

Exercice N°04 :

Écrire un programme C permettant de calculer le PGCD (algorithme d'Euclide) de deux nombres entiers A et B positifs.

Exercice N°05 :

Écrire un programme C qui calcule et affiche la Nième puissance d'un nombre réel X qui doit être lu et affiché en entrée. $X^N = X * X* ... * X$.

Puis compiler et exécuter le programme pour ***x=1.5 et N=3***

Exercice N°06 :

Écrire un programme en C où l'utilisateur doit deviner un **nombre secret**.

Il saisit un nombre dans la variable **CODE.**

Tant que code est différent du nombre secret, le programme indique :

- «code incorrect et Trop petit *! »* si code est inférieur
- «code incorrect et *Trop grand ! »* si code est supérieur

Dans le cas l'utilisateur trouve le bon nombre, le programme affiche :

« code correct ! »

TP Structure des Ordinateurs et Applications

Exercices supplémentaires – Série de TP N°04

Exercice- sup N°01 :

Écrire un algorithme / programme c qui permet d'effectuer une lecture conditionnée. Par exemple, on veut saisir un nombre entier ***N*** qui doit supérieur ou égale à 3. Si ***N < 3***, l'algorithme affiche une erreur et demande à nouveau de ré-saisir la valeur de ***N***

Exercice- sup N°02 :

Soit ***N*** un nombre entier strictement positif. Écrire un algorithme / Programme c qui permet d'afficher tousles diviseurs de ***N***. Exemple ***N = 8***, les diviseurs de ***N = {1, 2, 4, 8}***

Exercice- sup N°03 :

Soit ***N*** un nombre entier strictement positif. Écrire un algorithme / Programme c qui permet d'introduire la valeur de ***N*** et d'indiquer si ***N*** est premier ou non premier. Un nombre premier est un nombre qui possède uniquement deux diviseurs : ***1*** et lui-même.

Exercice- sup N°04 :

Écrire un algorithme / programme Pascal qui permet d'indiquer si entier strictement positif ***N*** est un nombreparfait ou non. Un nombre parfait est un nombre qui à la somme de ses diviseurs propres (L'ensemble desdiviseurs sauf lui-même). Exemple : ***N = 6***, les diviseurs propres sont : ***1, 2 et 3 et 6=1+2+3***, donc ***6*** est parfait.

Exercice- sup N°05 :

Écrire un programme en C qui demande à l'utilisateur d'entrer un nombre entier positif et qui utilise une boucle pour compter combien de chiffres contient ce nombre. Le programme doit ensuite afficher le résultat sous la forme « Ce nombre contient X chiffres ».

Exercice- sup N°06 :

Soit N un nombre entier strictement positif. Écrire un algorithme ou un programme en C qui permet d'introduire la valeur de N et de calculer la somme des nombres entiers de 1 jusqu'à N. Le programme doit afficher le résultat de cette somme et utiliser une boucle pour effectuer le calcul. Si N n'est pas strictement positif, le programme doit afficher un message d'erreur et ne pas effectuer le calcul.

Exercice- sup N°07 :

Écrire un programme en C qui demande à l'utilisateur d'entrer un nombre entier positif N et qui calcule et affiche la somme des nombres multiples de 3 ou de 5 compris entre 1 et N en utilisant une boucle.

TP Structure des Ordinateurs et Applications

Corrigé de la Série de TP N°4– *Les instructions itératives : Pour, Tant-que et Répéter*

Rappel :

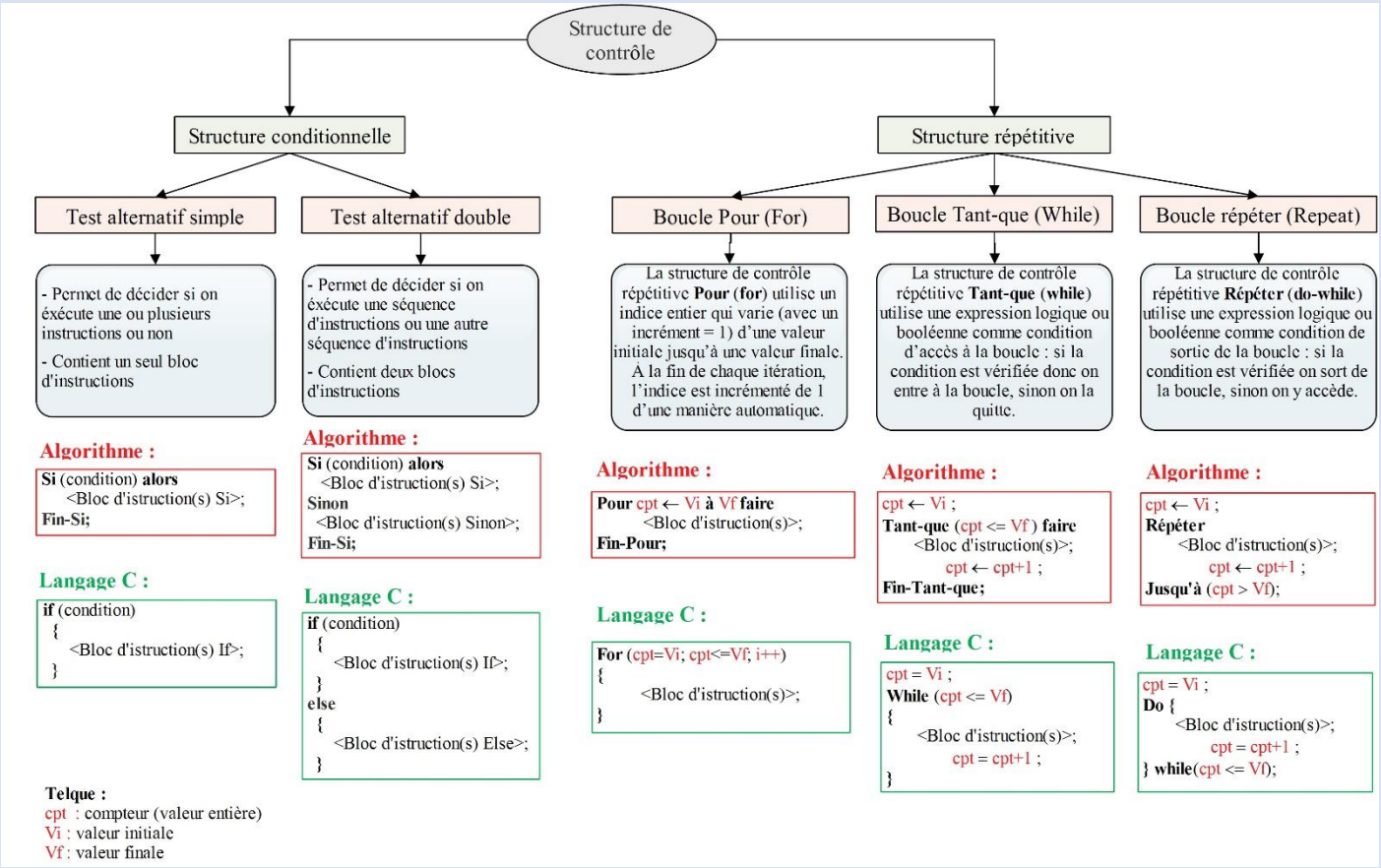
Structures de contrôle répétitives

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 (do-while)**

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** et 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 :

1. *Traduction de l'algorithme :*

Algorithme	Langages C
Algorithme exercice01 ; Variables N,i : Entier ; R,j : Réel ; Début Lire (N) ; R ←1; j ← 1; Pour i ← 1 à N faire R ← i / j * R ; j← i + 2 ; fin-pour ; écrire (‘R= ’,R) ; fin.	#include <stdio.h> int main() { int N, i; float R, j; printf("Entrez un entier N : "); scanf("%d", &N); R = 1; j = 1; for (i = 1; i <= N; i++) { R = i / j * R; j = i + 2; } printf("R = %f\n", R); return 0; }

2. *Dérouler l'algorithme pour N=3.*

Instructions	Variables				Affichage
	N	i	j	R	
Ecrire (‘donner un nombre N’) ;	-	-	-	-	donner un nombre N
Lire (N) ;	3	-	-	-	-
R ←1; j ← 1;	3		1	1	-
Pour i← 1 R ←i / j * R ; R ←1/1*1; R ←1*1; R ←1; j ←i + 2 ; j ← 1 + 2 ; j ← 3 ;	3	1	3	1	-
Pour i← 2 R ←i / j * R ; R ←2/3*1; R ←2/3; j ←i + 2 ; j ← 2 + 2 ; j ← 4 ;	3	2	4	0.67	-
Pour i← 3 R ←i / j * R ; R ←3/4*2/3; R ← 3/5; j ←i + 2 ; j ← 3 + 2 ; j ← 5 ;	3	3	5	0.50	-
écrire (‘R= ’,R) ;	3	3	5	0.50	R = 0.50

Compilation et exécution du programme pour N=3

main.c

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int N, i;
6      float R, j;
7      printf("Entrez un entier N : ");
8      scanf("%d", &N);
9      R = 1;
10     j = 1;
11     for (i = 1; i <= N; i++) {
12         R = i / j * R;
13         j = i + 2;
14     }
15     printf("R = %.2f\n", R);
16
17     return 0;
18 }
19
20
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\FFFF\bin\De...
Entrez un entier N : 3
R = 0.50

Process returned 0 (0x0)   execution time : 20.681 s
Press any key to continue.
```

1. Déduire l'expression générale de R.

$$R = \frac{2N!}{1.3.5\ldots(2N-1).(2N+1)}; OU \rightarrow R = \frac{2}{N+1}$$

2. Réécrire le programme C en remplaçant la boucle **pour** par la boucle **Tant-que** Réécrire le programme C en remplaçant la boucle **pour** par la boucle **tant-que** ensuite pas la boucle **répéter:**

Boucle Tant-que	Boucle Répéter
#include <stdio.h> int main() { int N, i; float R, j; printf("Entrez un entier N : "); scanf("%d", &N); R = 1; j = 1; i = 1; while (i <= N) { R = i / j * R; j = i + 2; i++; } printf("R = %.4f\n", R); return 0; }	#include <stdio.h> int main() { int N, i; float R, j; printf("Entrez un entier N : "); scanf("%d", &N); R = 1; j = 1; i = 1; if (N > 0) { // Vérifie que N est positif pour entrer dans la boucle do { R = i / j * R; j = i + 2; i++; } while (i <= N); } printf("R = %.4f\n", R); return 0; }

main.c

```c
#include <stdio.h>
int main() {
    int N, i;
    float R, j;
    printf("Entrez un entier N : ");
    scanf("%d", &N);
    R = 1;
    j = 1;
    i = 1;
    while (i <= N) {
        R = i / j * R;
        j = i + 2;
        i++;
    }
    printf("R = %.4f\n", R);
    return 0;
}
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\FFFF\bin...
Entrez un entier N : 3
R = 0.5000

Process returned 0 (0x0)   execution time : 2.255 s
Press any key to continue.
```

main.c

```c
#include <stdio.h>
int main() {
    int N, i;
    float R, j;
    printf("Entrez un entier N : ");
    scanf("%d", &N);
    R = 1;
    j = 1;
    i = 1;
    if (N > 0) { // Vérifie que N est positif
        do {
            R = i / j * R;
            j = i + 2;
            i++;
        } while (i <= N);
    }
    printf("R = %.4f\n", R);
    return 0;
}
```

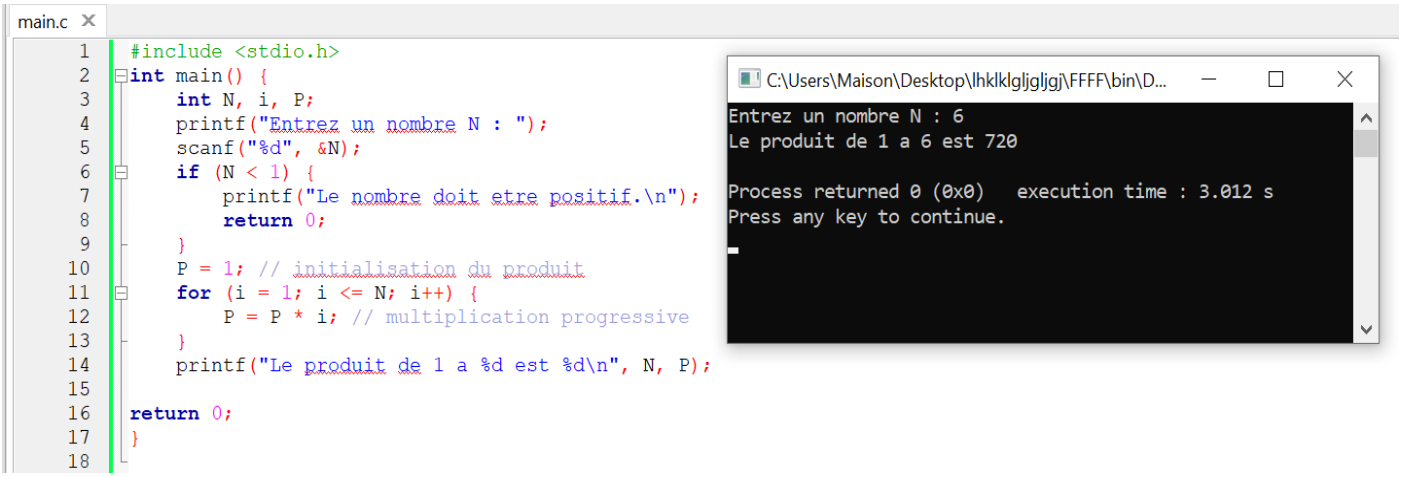
```
C:\Users\Maison\Desktop\lhklklgljgljgj\FFFF\...
Entrez un entier N : 3
R = 0.5000

Process returned 0 (0x0)   execution time : 2.404 s
Press any key to continue.
```

Exercice N°02 :

1/ Programme pour calculer l'expression P:

Algorithme	Programme en C
Algorithme ProdEntier ; Variables N, i, P : Entier ; Début ; Écrire "Entrez un nombre N" ; Lire N ; Si N < 1 alors ; Écrire "Le nombre doit être positif" ; Fin ; Sinon ; P ← 1 ; Pour i de 1 à N faire ; P ← P * i ; Fin Pour ; Écrire "Le produit de 1 à ", N, " est ", P ; Fin Si ; Fin.	#include <stdio.h> int main() { int N, i, P; printf("Entrez un nombre N : "); scanf("%d", &N); if (N < 1) { printf("Le nombre doit etre positif.\n"); return 0; } P = 1; // initialisation du produit for (i = 1; i <= N; i++) { P = P * i; // multiplication progressive } printf("Le produit de 1 a %d est %d\n", N, P); return 0; }

Compilation et exécution du programme pour N=6:



2/ Programme pour calculer l'expression S:

Algorithme	Programme en C
Algorithme SomPuiFact ; **Variables** N, i, F : Entier ; X, P, S : Réel ; **Début ;** Écrire "Entrez le nombre de termes N" ; Lire N ; Écrire "Entrez la valeur de X" ; Lire X ; // Initialisation S ← 0 ; P ← X ; F ← 1 ;	#include <stdio.h> int main() { int N, i, F; float X, P, S; // Entrée printf("Entrez le nombre de termes N : "); scanf("%d", &N); printf("Entrez la valeur de X : "); scanf("%f", &X); // Initialisation S = 0.0f; // somme P = X; // X^(2i+1) initial F = 1; // (2i+1)! initial

// Calcul terme par terme Pour i ← 0 à N-1 faire ; S ← S + P / F ; P ← P * X * X ; // mise à jour de X^(2i+1) F ← F * (2*i* + 2) * (2i + 3) ; // mise à jour de (2i+1)! Fin Pour ; Écrire "La somme S = ", S ; **Fin.**	// Calcul terme par terme for (i = 0; i < N; i++) { S = S + P / F; // ajouter le terme P = P * X * X; // mettre à jour la puissance : X^(2(i+1)+1) F = F * (2*i + 2) * (2*i + 3); // mettre à jour la factorielle : (2(i+1)+1)! } // Affichage du résultat printf("La somme S = %.4f\n", S); return 0; }

Compilation et exécution du programme pour X=2 et N=3:

main.c

```c
#include <stdio.h>
int main() {
    int N, i, F;
    float X, P, S;
    // Entrée
    printf("Entrez le nombre de termes N : ");
    scanf("%d", &N);
    printf("Entrez la valeur de X : ");
    scanf("%f", &X);
    // Initialisation
    S = 0.0f;     // somme
    P = X;        // X^(2i+1) initial
    F = 1;        // (2i+1)! initial
    // Calcul terme par terme
    for (i = 0; i < N; i++) {
        S = S + P / F;     // ajouter le terme
        P = P * X * X;    // mettre à jour la puissance : X^(2(i+1)+1)
        F = F * (2*i + 2) * (2*i + 3);  // mettre à jour la factorielle : (2(i+1)+1)!
    }
    // Affichage du résultat
    printf("La somme S = %.4f\n", S);
    return 0;
}
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\FFFF\bin\Debug\FF...
Entrez le nombre de termes N : 3
Entrez la valeur de X : 2
La somme S = 3.6000

Process returned 0 (0x0)   execution time : 6.183 s
Press any key to continue.
```

Exercice N°3 :

Algorithme	Programme en C
Algorithme Somme_Produit ; Variables N, i : Entier ; somme : Entier ; produit : long ; Début ; Écrire("Entrez un entier N") ; Lire (N) ; // Initialisation somme ← 0 ; produit ← 1 ; // Calcul de la somme de 0 à N Pour i ← 0 à N faire somme ← somme + i ; Fin Pour ; // Calcul du produit de 1 à N Pour i ← 1 à N faire produit ← produit * i ; Fin Pour ; // Affichage des résultats Écrire ("La somme des nombres de 0 à ", N, " est : ", somme) ; Écrire ("Le produit des nombres de 1 à ", N, " est : ", produit) ; Fin.	#include <stdio.h> int main() { int N, i; int somme = 0; long produit = 1; // Utiliser long pour éviter le dépassement // Entrée printf("Entrez un entier N : "); scanf("%d", &N); // Calcul de la somme et du produit for (i = 0; i <= N; i++) { somme += i; // somme de 0 à N } for (i = 1; i <= N; i++) { produit *= i; // produit de 1 à N } // Affichage des résultats printf("La somme des nombres de 0 à %d est : %d\n", N, somme); printf("Le produit des nombres de 1 à %d est : %ld\n", N, produit); return 0;

	}

Compilation et exécution du programme pour N=15:

main.c

```c
#include <stdio.h>
int main() {
    int N, i;
    int somme = 0;
    long produit = 1;  // Utiliser long pour éviter le dépassement
    // Entrée
    printf("Entrez un entier N : ");
    scanf("%d", &N);

    // Calcul de la somme et du produit
    for (i = 0; i <= N; i++) {
        somme += i;         // somme de 0 à N
    }
    for (i = 1; i <= N; i++) {
        produit *= i;       // produit de 1 à N
    }
// Affichage des résultats
    printf("La somme des nombres de 0 a %d est : %d\n", N, somme);
    printf("Le produit des nombres de 1 a %d est : %ld\n", N, produit);
    return 0;
}
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\FFFF\bin\...
Entrez un entier N : 15
La somme des nombres de 0 a 15 est : 120
Le produit des nombres de 1 a 15 est : 2004310016

Process returned 0 (0x0)   execution time : 4.327 s
Press any key to continue.
```

Exercice N°4 :

Algorithme	Programme en C
Algorithme PGCD; Variables A, B, P : Entier ; Début ; Écrire("Donner A et B :") ; Lire(A, B) ; Tant que A ≠ B faire ; Si A > B alors ; A ← A - B ; Sinon ; B ← B - A ; Fin Si ; Fin Tant que ; P ← A ; // ou B, ils sont égaux Écrire("Le PGCD est : ", P) ; Fin ;	#include <stdio.h> int main() { int A, B, P; printf("Donner A et B : "); scanf("%d%d", &A, &B); while (A != B) { if (A > B) { A = A - B; } else { B = B - A; } } P = A; // ou B, les deux sont égaux ici printf("Le PGCD est : %d\n", P); return 0; }

Compilation et exécution du programme :

main.c

```c
#include <stdio.h>
int main()
{
    int A, B, P;
    printf("Donner A et B : ");
    scanf("%d%d", &A, &B);
    while (A != B)
    {
        if (A > B)
        {
            A = A - B;
        }
        else
        {
            B = B - A;
        }
    }
    P = A;  // ou B, les deux sont égaux ici
    printf("Le PGCD est : %d\n", P);
    return 0;
}
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\FFFF\bin\Debug\...
Donner A et B : 5 9
Le PGCD est : 1

Process returned 0 (0x0)   execution time : 5.124 s
Press any key to continue.
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\FFFF\bin\Debu...
Donner A et B : 4 6
Le PGCD est : 2

Process returned 0 (0x0)   execution time : 3.758 s
Press any key to continue.
```

Exercice N°5 :

Algorithme	Programme en C
Algorithme Puissance_Nieme ; Variables X : Réel ; N, i : Entier ; P : Réel ; Début ; Écrire("Entrez le nombre réel X :") ; Lire(X) ; Écrire("Entrez l'entier N :") ; Lire(N) ; // Initialisation P ← 1 ; // Calcul de X^N Pour i ← 1 à N faire ; P ← P * X ; Fin Pour ; Écrire("La puissance X^N est : ", P) ; Fin ;	#include <stdio.h> int main() { double X, P; int N, i; // Entrée printf("Entrez le nombre réel X : "); scanf("%lf", &X); printf("Entrez l'entier N : "); scanf("%d", &N); // Initialisation P = 1; // Calcul de X^N for(i = 1; i <= N; i++) { P = P * X; } // Affichage printf("La puissance %.2lf^%d est : %.2lf\n", X, N, P); return 0; }

Compilation et exécution du programme pour x=1.5 et N=3 :

main.c

```c
#include <stdio.h>

int main() {
    double X, P;
    int N, i;
    // Entrée
    printf("Entrez le nombre reel X : ");
    scanf("%lf", &X);
    printf("Entrez l'entier N : ");
    scanf("%d", &N);
    // Initialisation
    P = 1;
    // Calcul de X^N
    for(i = 1; i <= N; i++) {
        P = P * X;
    }
    // Affichage
    printf("La puissance %.2lf^%d est : %.2lf\n", X, N, P);
    return 0;
}
```

```
C:\Users\Maison\Desktop\lhklklgljgljgj\FFFF\bin\Deb...
Entrez le nombre reel X : 1.5
Entrez l'entier N : 3
La puissance 1.50^3 est : 3.38

Process returned 0 (0x0)   execution time : 12.494 s
Press any key to continue.
```

Exercice N°6 :

Programme en c

```
#include <stdio.h>
int main() {
   int CODE;
   int secret = 15426; // Nombre secret
   printf("Devinez le nombre secret :\n");
   do {
      printf("Entrez votre code : ");
      scanf("%d", &CODE);
      if (CODE < secret) {
         printf("code incorrect et Trop petit !\n");
      } else if (CODE > secret) {
         printf("code incorrect et Trop grand !\n");
      } else {
         printf("Code correct !\n");
      }
   } while (CODE != secret); // Tant que le code n'est pas correct
   return 0;
}
```

Compilation et exécution du programme