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

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Année universitaire 2024/2025
1ère Année Ingénieur

TP Structure des ordinateurs et applications

Corrigé de la série de TP N°3 – *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)

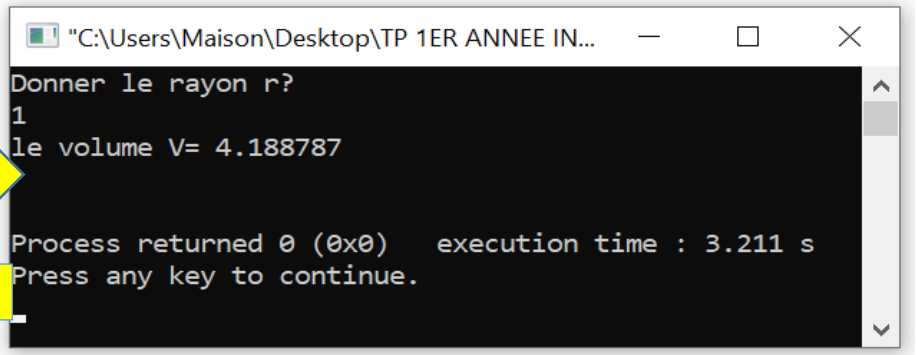
1) Traduction de l'algorithme en Programme C

Algorithme	**Programme C**

Algorithme Exo1;
Constantes
Pi=3.14 ;
Variables
r, S, V : **Réel**;
Début
//Entrées
Écrire("Donner le rayon r : ");
Lire(r);
//Traitement
S ← 4*Pi*r*r;
V ← S*r/3;
//Sortie
Écrire("le volume V= ",V);
Fin.
	#include <stdio.h> *// déclaration de la Bibliothèque*
	
	int main( )
	{
	const float Pi=3.14159; *//déclaration des constantes*
	float r, S, V; *//déclaration des variables*
	// Déclaration des entrées
	printf("Donner le rayon r?\n");
	scanf("%f",&r);
	// Traitement
	S = 4*Pi*r*r;
	V = (S*r)/3;
	// Affichage des résultats
	printf("le volume V=%f \n\n", V);
	
	return 0;
	}

2) Compilation et exécution du programme pour : r = 1

```c
#include <stdio.h> //déclaration de la bibliothèque (entrée & sorties)

int main()
{
    const float Pi=3.14159;      //déclaration des constantes
    float r, S, V;               //déclaration des variables
     // Déclaration des entrées
    printf("Donner le rayon r?\n");
    scanf("%f",&r);
      // Traitement
    S = 4*Pi*r*r;
    V = (S*r)/3;
     // Affichage des résultats
    printf("le volume V= %f \n\n",V);

    return 0;
}
```



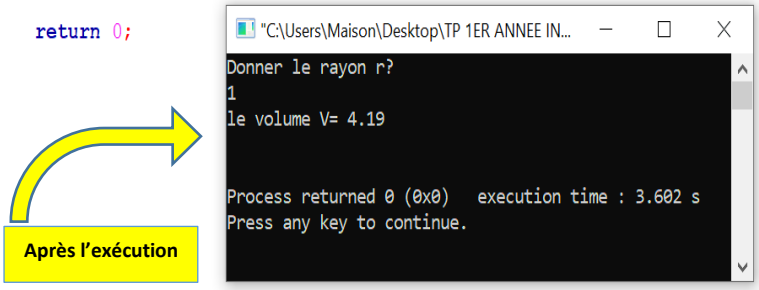
Après l'exécution

3) Remplaçant dans la dernière instruction `%f` par `%.2f`, puis réexécutant le programme :

Programme C

```c
#include <stdio.h>

int main()
{
   const float Pi=3.14159;  //déclaration des constantes
   float r, S, V;           //déclaration des variables
    // Déclaration des entrées
   printf("Donner le rayon r?\n");
   scanf("%f",&r);
     // Traitement
   S = 4*Pi*r*r;
   V = (S*r)/3;
    // Affichage des résultats
   printf("le volume V= %.2f    \n\n",V);

   return 0;
}
```

```c
#include <stdio.h> //déclaration de la bibliothèque (entrée & sorties)

int main()
{
    const float Pi=3.14159;      //déclaration des constantes
    float r, S, V;               //déclaration des variables
     // Déclaration des entrées
    printf("Donner le rayon r?\n");
    scanf("%f",&r);
      // Traitement
    S = 4*Pi*r*r;
    V = (S*r)/3;
     // Affichage des résultats
    printf("le volume V= %.2f \n\n",V);

    return 0;
}
```



Après l'exécution

`printf("le volume V= %.2f    \n\n",V);` Où .2 représente deux chiffres après la virgule.

4) *Ajoutant une nouvelle instruction au programme pour afficher le résultat de la surface S :*

Programme C

```c
#include <stdio.h>

int main()
{
   const float Pi=3.14159;  //déclaration des constantes
   float r, S, V;           //déclaration des variables
    // Déclaration des entrées
   printf("Donner le rayon r?\n");
   scanf("%f",&r);
    // Traitement
   S = 4*Pi*r*r;
   V = (S*r)/3;
    // Affichage des résultats
   printf("le volume V= %.2f     \n\n",V);
   printf("la surface  S= %.2f     \n\n",S);
   return 0;
}
```

```c
#include <stdio.h> //déclaration de la bibliothèque (entrée & sorties)

int main()
{
    const float Pi=3.14159;      //déclaration des constantes
    float r, S, V;               //déclaration des variables
     // Déclaration des entrées
    printf("Donner le rayon r?\n");
    scanf("%f",&r);
      // Traitement
    S = 4*Pi*r*r;
    V = (S*r)/3;
     // Affichage des résultats
    printf("le volume V= %.2f \n\n",V);
    printf("la surface S= %.2f \n\n",S);

    return 0;
}
```

```
"C:\Users\Maison\Desktop\TP 1ER ANNEE I...
Donner le rayon r?
1
le volume V= 4.19

la surface S= 12.57


Process returned 0 (0x0)   execution time : 3.185 s
ress any key to continue.
```

Après l'exécution

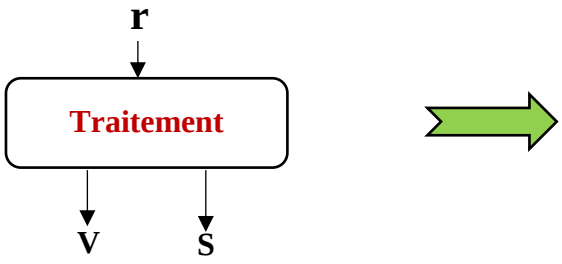
5) *Déroulement de l'algorithme pour R=1 :*

Instructions	Variables			Affichage
	r	S	V	
Écrire("Donner le rayon r : ")	/	/	/	Donner le rayon r :
Lire (r)	1	/	/	
S ← 4*Pi*r*r;	1	12.57	/	
V ← S*r/3;	1	12.57	4.19	
Écrire("Le volume V= ",V);	1	12.57	4.19	Le volume V= 4.19
Écrire("La surface S= ",S);	1	12.57	4.19	La surface S= 12.57

6) *Déduction de ce que fait le programme :*

Le programme calcule le volume et la surface d'une sphère.

7) *Le schéma complet est :*



Entrée :

Lire (r) ;

Traitements :

S ← 4*Pi*r*r;

V ← S*r/3;

Sorties :

Écrire (V) ;
Écrire (S) ;

Chaque algorithme possède des variables d'entrée, des variables de sorties, constantes et une partie du traitement :

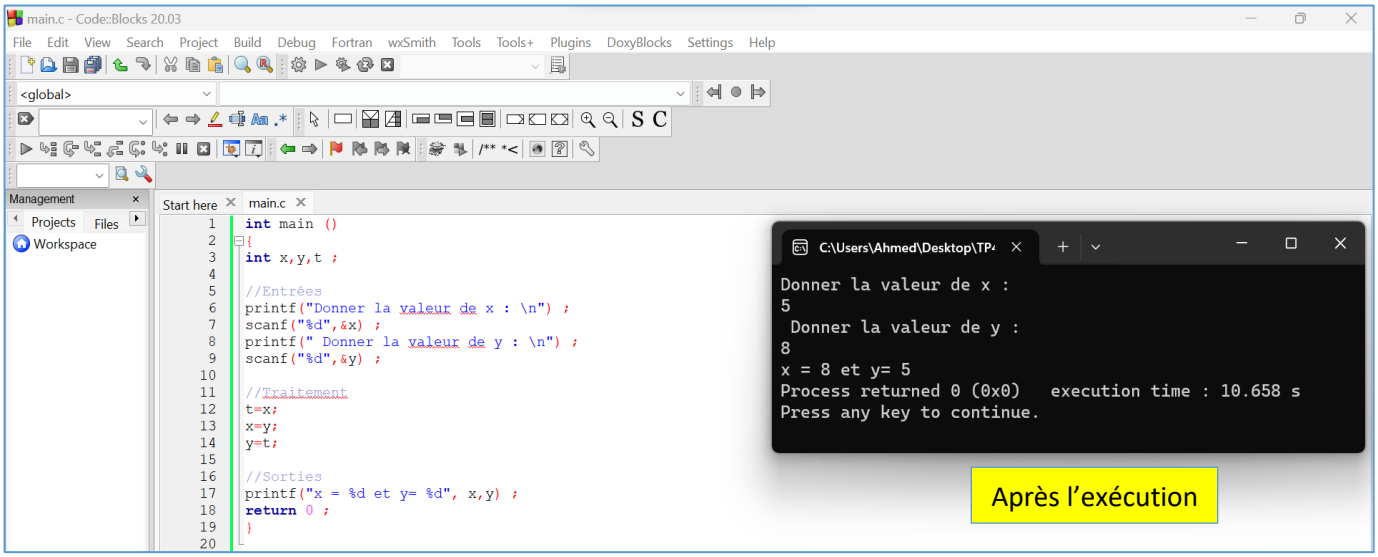
- Les variables d'entrée sont les variables lues (en utilisant l'instruction **Lire**) ;
- Les variables de sorties sont les variables affichées (en utilisant l'instruction **Ecrire**) ;
- Les données intermédiaires qui peuvent être des variables ou des constantes (dans notre cas **Pi**) qui sont des données non lues et non écrites utilisées pendant le traitement ;
- Traitement : contient les instructions d'affectation, tests et les boucles.

Exercice N°02 : *(Enoncé du problème ➔ Algorithme ➔ Programme C)*

1) Permutation entre les deux variables X et Y :

Algorithme	**Programme C**

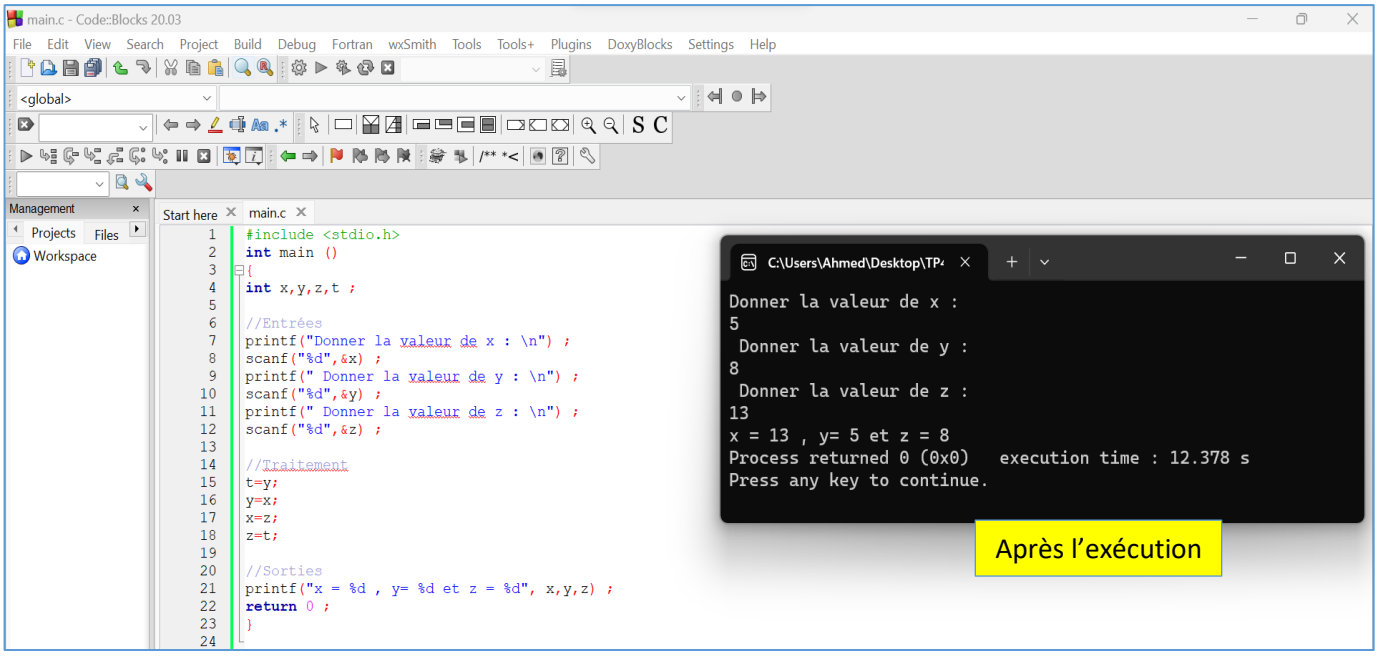
Algorithme Exo2_1;
Variables
x, y, t : **entier**;
Début
//Entrées
Ecrire("Donner la valeur de x : ") ;
Lire(x) ;
Ecrire("Donner la valeur de y : ") ;
Lire(y) ;
//Traitement
t ← x;
x ← y;
y ← t ;
//Sorties
Écrire("x=", x, "y=", y) ;
Fin.
	`#include <stdio.h>`
	`int main( )`
	`{`
	`int x,y,t ;`
	`//Entrées`
	`printf("Donner la valeur de x : \n") ;`
	`scanf("%d",&x) ;`
	`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 :

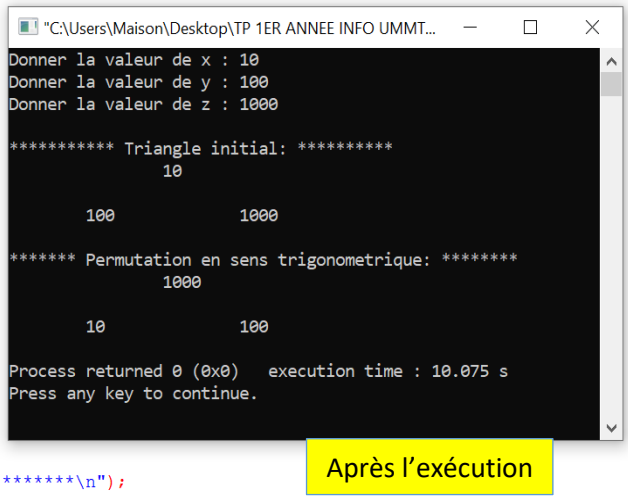
Algorithme	**Programme C**

Algorithme Exo2_2;
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;
	}



3) Affichage de trois variables L, U et V sous forme d'un triangle, puis permutation de ces variables dans le sens trigonométrique :

```c
#include <stdio.h>

int main() {
    int x, y, z, t;
    // Saisie des valeurs
    printf("Donner la valeur de x : ");
    scanf("%d", &x);
    printf("Donner la valeur de y : ");
    scanf("%d", &y);
    printf("Donner la valeur de z : ");
    scanf("%d", &z);

    // Affichage initial
    printf("\n********** Triangle initial: **********\n");
    printf("\t\t%d\n\n", x);
    printf("\t%d\t\t%d\n", y, z);

    // Permutation dans le sens trigonométrique
    t = x; // Stocker x dans t
    x = z; // x prend la valeur de z
    z = y; // z prend la valeur de y
    y = t; // y prend la valeur initiale de x

    // Affichage après permutation
    printf("\n******* Permutation en sens trigonometrique: ********\n");
    printf("\t\t%d\n\n\t%d\t\t%d\n", x, y, z);
    return 0;
}
```

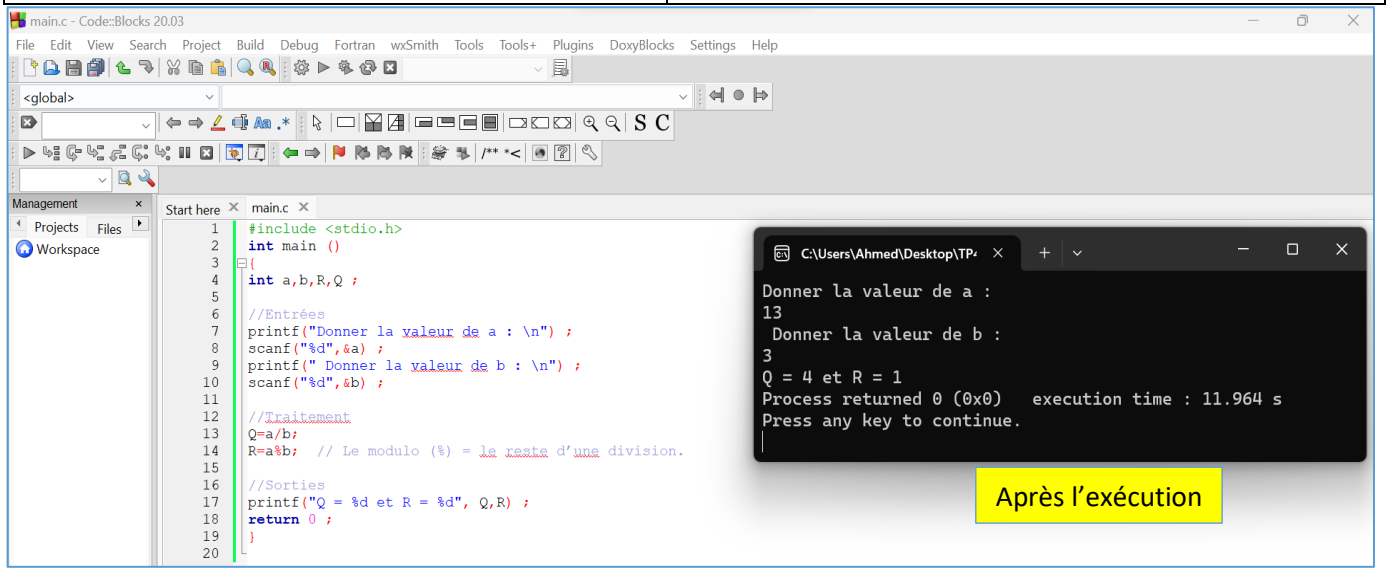


Après l'exécution

4) Calcule de quotient et le reste de la division euclidienne de a par b :

Algorithme	**Programme C**

Algorithme Exo2_4;
Variables
a, b, Q, R : **entier**;
Début
//Entrées
Ecrire("Donner la valeur de a : ") ;
Lire(a) ;
Ecrire("Donner la valeur de b : ") ;
Lire(b) ;
//Traitement
Q ← a div b; //div : division entière
R ← a mod b; //mod : reste de division
//Sorties
Écrire("Q=", Q, " et R= ", R) ;
Fin.
	#include <stdio.h>
	int main( )
	{
	int a,b,R,Q ;
	//Entrées
	printf("Donner la valeur de a : \n") ;
	scanf("%d",&a) ;
	printf(" Donner la valeur de b : \n") ;
	scanf("%d",&b) ;
	//Traitement
	Q=a/b;
	R=a%b; // Le modulo (%) = le reste d'une division.
	//Sorties
	printf("Q = %d et R = %d", Q,R) ;
	return 0;
	}

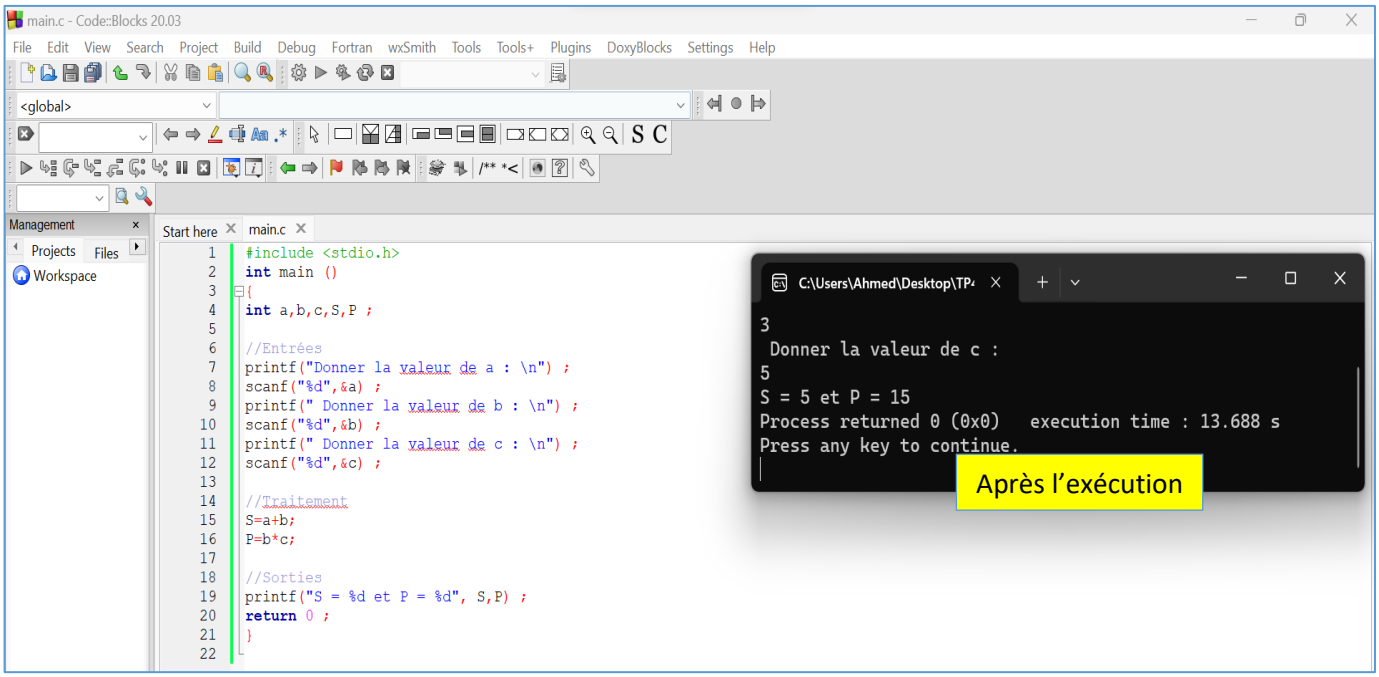


Après l'exécution

5) Calcule de la somme de a et b et le produit de b et c :

Algorithme	Programme C

Algorithme Exo2_5;
Variables
a, b, c, S, P : **entier**;
 	
Début
//Entrées
Ecrire("Donner la valeur de a : ") ;
Lire(a) ;
Ecrire("Donner la valeur de b : ") ;
Lire(b) ;
Ecrire("Donner la valeur de c : ") ;
Lire(c) ;
 	
//Traitement
S ← a + b;
P ← b * c;
 	
//Sorties
Écrire("S=", S, "P= ", P) ;
Fin.
	#include <stdio.h>
	int main( )
	{
	int a,b,c,S,P ;
	
	//Entrées
	printf("Donner la valeur de a : \n") ;
	scanf("%d",&a) ;
	printf(" Donner la valeur de b : \n") ;
	scanf("%d",&b) ;
	printf(" Donner la valeur de c : \n") ;
	scanf("%d",&c) ;
	
	//Traitement
	S=a+b;
	P=b*c;
	
	//Sorties
	printf("S = %d et P = %d", S,P) ;
	return 0;
	}



6) Calcule de la valeur absolue et le carré d'un nombre :

Algorithme	Programme C

Algorithme Exo2_6;
Variables
x : **réel**;
va, ca : **entier**;
 	
Début
//Entrées
Ecrire("Donner la valeur de x : ") ;
Lire(x) ;
 	
//Traitement
va ← |x|;
ca ← x^2;
 	
//Sorties
Écrire("va=", va, "ca=", ca) ;
Fin.
	`#include <stdio.h>`
	`#include<math.h>`
	`int main( )`
	`{`
	`float x ; int va, ca ;`
	`//Entrées`
	`printf("Donner la valeur de x : \n") ;`
	`scanf("%f",&x) ;`
	`//Traitement`
	`va=abs(x); /*Cette fonction retourne la valeur absolue d'un nombre (Nb : renvoie un nombre entier)*/`
	`// ca=x*x ; ou bien :`
	`ca=pow(x,2); /*Cette fonction permet de calculer la puissance d'un nombre */`
	`//Sorties`
	`printf("va = %d et ca = %d", va, ca) ;`
	`return 0;`
	`}`

- ***L'importance du choix des types de variables :***

☞ ***Le résultat retourné en entier***

```c
#include <stdio.h>
#include<math.h>
int main( )
{
float x ;
int va, ca ;
//Entrées
printf("Donner la valeur de x : \n") ;
scanf("%f",&x) ;
//Traitement
va=abs(x);    /*Cette fonction retourne la valeur absolue d'un nombre (Nb : renvoie un nombre entier)*/
//ca=x*x; ou bien :
ca=pow(x,2);    /*Cette fonction POW permet de calculer la puissance d'un nombre */
//Sorties
printf("va = %d et ca = %d", va, ca) ;
return 0;
}
```

```
"C:\Users\Maison\Desktop\TP 1ER ANNEE INFO U...
-6.5
va = 6 et ca = 42
Process returned 0 (0x0)   execution time : 7.995 s
Press any key to continue.
```

Après l'exécution

☞ ***Le résultat retourné en réel***

```c
#include <stdio.h>
#include <math.h>

int main() {
    float x;
    float va, ca; // ca et va en réels
    // Entrées
    printf("Donner la valeur de x : \n");
    scanf("%f", &x);
    // Traitement
    va = fabsf(x); // fabsf nous retourne float
    ca = powf(x, 2); // utilisant powf pour calculer power en float
    // Sorties
    printf("va = %.2f et ca = %.2f\n", va, ca); // utilisant %.2f pour 2 chiffres après la virgule
    return 0;
}
```

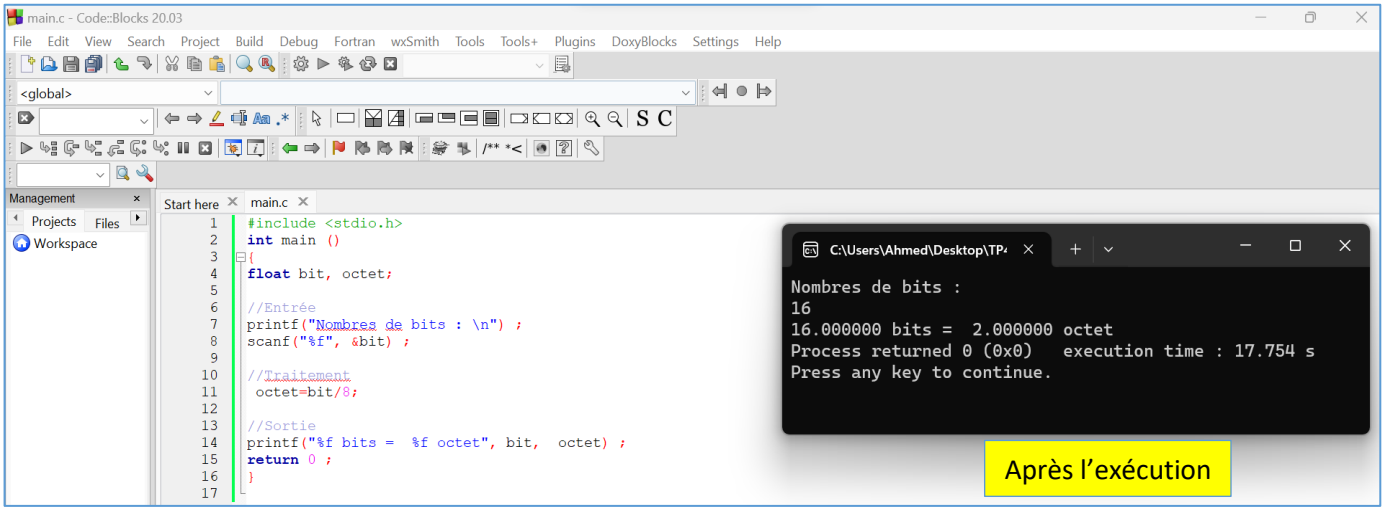
```
"C:\Users\Maison\Desktop\TP 1ER ANNEE INFO UMMTO\SEMESTRE 1\TP-3\PROGRA...
Donner la valeur de x :
-6.5
va = 6.50 et ca = 42.25

Process returned 0 (0x0)   execution time : 7.164 s
Press any key to continue.
```

Après l'exécution

7) Conversion en octets un nombre donné en bits :

Algorithme	Programme C

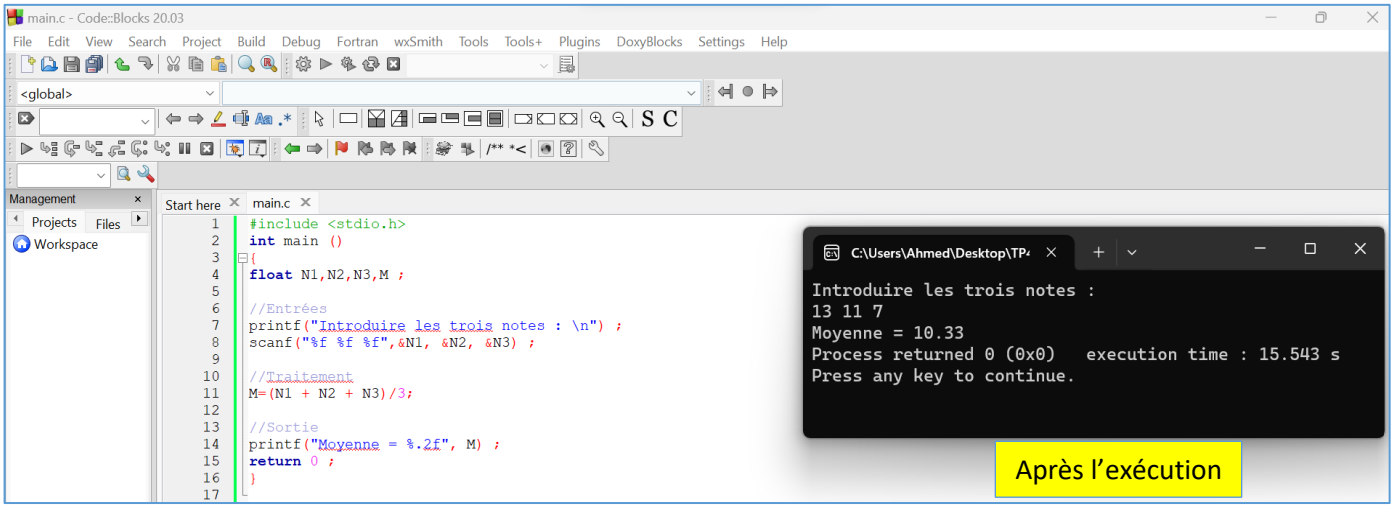
Algorithme Exo2_7;
Variables
bit, octet : **réel**;
Début
//Entrée
Écrire("Nombres de bits =") ;
Lire(bit) ;
//Traitement
octet ← bit/8;
//Sortie
Écrire(bit, " bits =", octet, " octet");
Fin.
	#include <stdio.h>
	int main( )
	{
	float bit, octet;
	//Entrée
	printf("Nombres de bits : \n") ;
	scanf("%f", &bit) ;
	//Traitement
	octet=bit/8;
	//Sortie
	printf("%f bits = %f octet", bit, octet) ;
	return 0;
	}



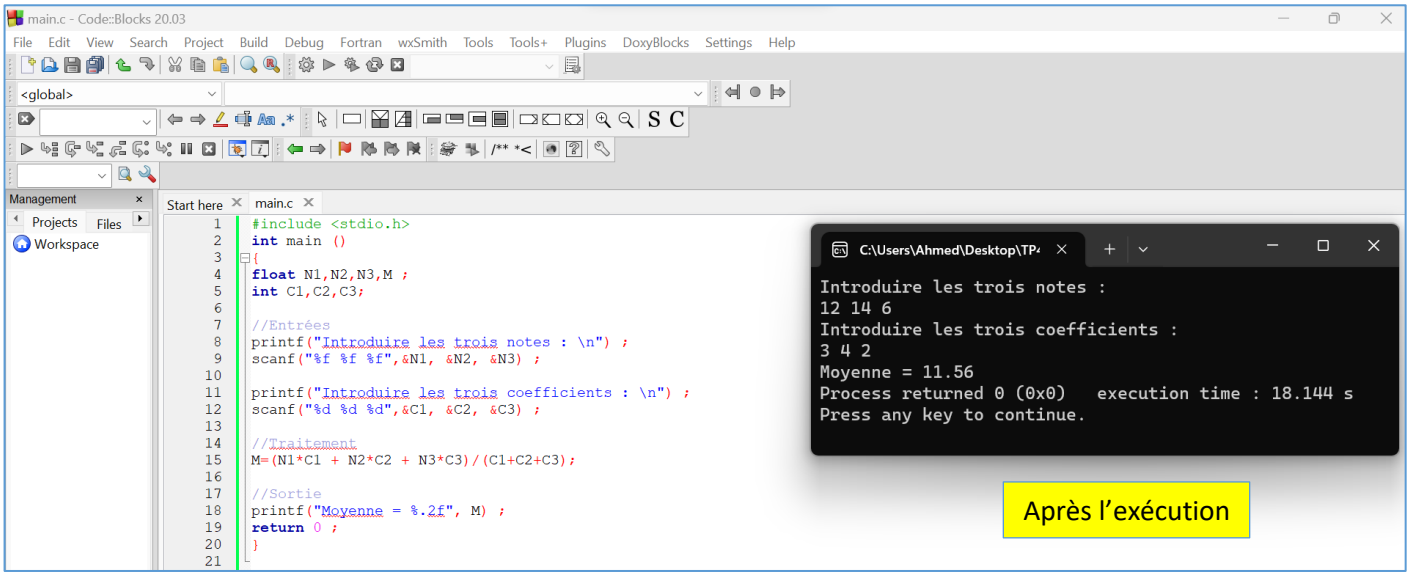
8.a) Lecture des notes de trois matières (N1, N2 et N3), puis calcul et affichage de leur moyenne M :

Algorithme	Programme C

Algorithme Exo2_8_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;
	}



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

Algorithme	**Programme C**

Algorithme Exo2_8_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( )
	{
	float N1,N2,N3,M ;
	int C1,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;
	}