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

Année universitaire 2026/2027
1ère Année Technologie

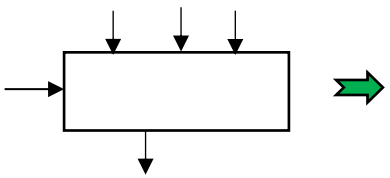
TP – Structure des Ordinateurs et applications

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

Exercice N°01 :

Soit l'algorithme suivant :

Algorithme	Questions

Algorithme Exo1**;**
Variables
N : Entier ;
P, PT : Réels ;
Début
// Entrées
Écrire("Donner le nombre d'articles :") ;
Lire(N) ;
Écrire("Donner le prix unitaire :") ;
Lire(P) ;
// Traitement
PT ← N × P ;
// Sortie
Écrire ("Le prix total est : ", PT) ;
Fin.
	1. Lister les variables de l'algorithme et préciser leurs types en C.
	2. Traduire l'algorithme en programme C.
	3. Compiler et exécuter le programme avec (***N=7***) et (***P=12.5***).
	4. Écrire les instructions de lecture de (***N***) et (***P***) en ***C*** :
	• **a)** Avec un `scanf()` pour chaque variable.
	• **b)** Avec un seul `scanf()` pour les deux variables.
	5. Écrire les instructions d'affichage du prix total `PT` avec `printf()` :
	• **a)** Afficher uniquement la valeur de `PT`.
	• **b)** Afficher un message explicatif suivi de `PT`.
	6. Modifier la partie affichage de votre programme en remplaçant l'instruction actuelle par les quatre instructions suivantes :
	1. printf("Prix total sans formatage :%f\n", PT);
	2. printf("Prix total avec formatage :%8.2f\n", PT);
	3. printf("Prix total avec formatage :%5.2f\n", PT);
	4. printf("Prix total avec formatage :%.2f\n", PT);
	7. Compiler et exécuter le programme avec (***N=7***) et (***P=12.5***), que remarquez-vous concernant les résultats affichés ? Expliquez la différence entre les différents formats d'affichage utilisés.
	8. Compléter le schéma suivant :
	
	Entrée :
	Traitements :
	Sorties :

Corrigé :

1. Variables et types en C

Variable	Type algorithmique	Type en C
`N`	Entier	`int`
`P`	Réel	`float`
`PT`	Réel	`float`

2. Traduction en C

```c
#include <stdio.h>
int main()
{
    int N;
    float P, PT;

    // Entrées
    printf("Donner le nombre d'articles : ");
    scanf("%d", &N);

    printf("Donner le prix unitaire : ");
    scanf("%f", &P);

    // Traitement
    PT = N * P;

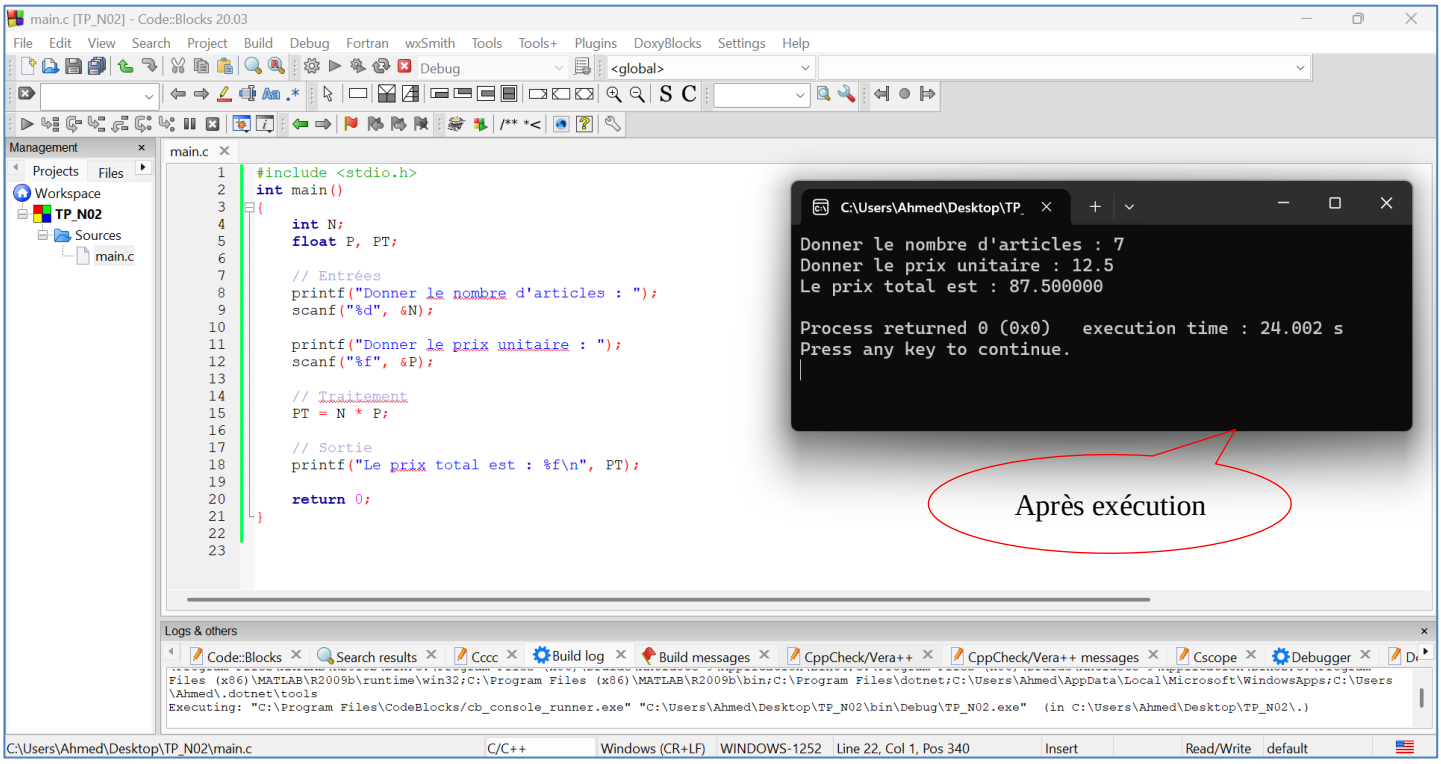
    // Sortie
    printf("Le prix total est : %f\n", PT);

    return 0;
}
```

►Lien du programme :

https://onlinegdb.com/GAttv1spK

3. Exécution pour N = 7 et P = 12.5



4. Lecture des données

a) Une instruction `scanf()` par variable :

```
scanf("%d", &N);
scanf("%f", &P);
```

b) Une seule instruction `scanf()` pour les deux variables :

```
scanf("%d %f", &N, &P);
```

5. Affichage du prix total

a) Uniquement la valeur :

```
printf("%f", PT);
```

b) Message explicatif suivi de la valeur :

```
printf("Le prix total est : %f", PT);
```

6. Formats d'affichage (PT = 87.5)

Instruction	Affichage obtenu (_ = espace)
`%f`	`87.500000`
`%8.2f`	`___87.50`
`%5.2f`	`87.50`
`%.2f`	`87.50`

Ce que l'on remarque

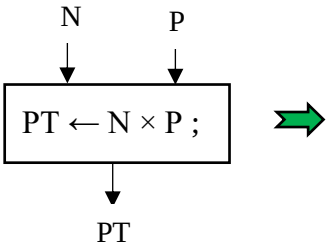
- Le format `%f` affiche **6 chiffres après la virgule** par défaut : `87.500000`.
- Les trois autres formats limitent l'affichage à **2 décimales** : `87.50`.
- Seul `%8.2f` ajoute des espaces devant le nombre. Les formats `%5.2f` et `%.2f` donnent le même affichage ici.

Explication des formats

La forme générale est **`%L.Pf`**, où **`L`** est la largeur minimale du champ (nombre total de caractères, point décimal compris) et **`P`** est le nombre de chiffres après la virgule.

Format	Signification	Résultat
`%f`	précision par défaut (6 décimales), pas de largeur imposée	`87.500000`
`%8.2f`	2 décimales, largeur minimale 8. `87.50` occupe 5 caractères, donc 3 espaces sont ajoutés à gauche	`___87.50`
`%5.2f`	2 décimales, largeur minimale 5. `87.50` occupe exactement 5 caractères, donc aucun espace n'est ajouté	`87.50`
`%.2f`	2 décimales, largeur libre (juste la place nécessaire)	`87.50`

7. Schéma à compléter



Entrée :

```
Lire (N) ;
Lire (P) ;
```

Traitement :

```
P ← N × P ;
```

Sorties :

```
Écrire ("Le prix total est :  ", PT) ;
```

Exercice N°02 :

Pour chacun des problèmes suivants, écrire son algorithme, puis le traduire en programme C:

1) Permuter entre les deux variables X et Y?
2) Permuter 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 ?
3) Calculer la valeur absolue *VA*, le carré *CA* et la racine carrée *RC* d'un nombre entier *N* ?
4) Calculer le quotient et le reste de la division euclidienne de *A* par *B*?

Corrigé :

1) Permuter X et Y

Idée : il faut une variable intermédiaire `Z`. Écrire `X ← Y` puis `Y ← X` perdrait l'ancienne valeur de X.

Algorithme	**Programme C**

Algorithme Permut1 ;
Variables
X, Y, Z : Entiers ;
Début
Écrire("Donner X et Y :") ;
Lire(X, Y) ;
Z ← X ;
X ← Y ;
Y ← Z ;
Écrire("Après permutation : X = ", X, " Y = ", Y) ;
Fin.
	#include <stdio.h>
	int main()
	{
	int X, Y, Z;
	printf("Donner X et Y : ");
	scanf("%d %d", &X, &Y);
	Z = X;
	X = Y;
	Y = Z;
	printf("Apres permutation : X = %d, Y = %d\n", X, Y);
	return 0;
	}

►**Lien du programme**

https://onlinegdb.com/aidryPH3r9

2) Permutation circulaire de X, Y et Z

Algorithme	**Programme C**

Algorithme Permut2 ;
Variables
X, Y, Z, H : Entiers ;
Début
Écrire("Donner X, Y et Z :") ;
Lire(X, Y, Z) ;
H ← Z ;
Z ← Y ;
Y ← X ;
X ← H ;
Écrire("Après permutation : X = ", X, " Y = ", Y, "
Z = ", Z) ;
Fin.
	#include <stdio.h>
	int main()
	{
	int X, Y, Z, H;
	
	printf("Donner X, Y et Z : ");
	scanf("%d %d %d", &X, &Y, &Z);
	
	H = Z;
	Z = Y;
	Y = X;
	X = H;
	printf("Apres permutation : X = %d, Y = %d, Z =
	%d\n", X, Y, Z);
	return 0;
	}

►**Lien du programme**

https://onlinegdb.com/Y5r_NrOSn

3) Carré CA et racine carrée RC d'un entier positif N

Algorithme	**Programme C**
Algorithme Calculs ; **Variables** N, CA : Entiers ; RC : Réel ; **Début** Écrire("Donner un entier positif N :") ;	**#include <stdio.h>** **#include <math.h>** **int main()** **{** int N, CA; float RC;

Lire(N) ;
CA ← N^2 ;
RC ← $\sqrt{N}$;
Écrire("CA = ", CA) ;
Écrire("RC = ", RC) ;
Fin.
	printf("Donner un entier positif N : ");
	scanf("%d", &N);
	
	CA = N * N; ou CA=pow(N,2) ;
	RC = sqrt(N);
	
	printf("Carre CA = %d\n", CA);
	printf("Racine carree RC = %.2f\n", RC);
	
	return 0;
	}

►**Lien du programme**

https://onlinegdb.com/l-cZctIaU

4) Quotient et reste de la division euclidienne de A par B (A et B > 0)

Algorithme	**Programme C**

Algorithme Division ;
Variables
A, B, Q, R : Entiers ;
Début
Écrire("Donner A et B (A > 0, B > 0) :") ;
Lire(A, B) ;
Q ← A div B ;
R ← A mod B ;
Écrire("Quotient = ", Q) ;
Écrire("Reste = ", R) ;
Fin.
	#include <stdio.h>
	int main()
	{
	int A, B, Q, R;
	
	printf("Donner A et B (A > 0, B > 0) : ");
	scanf("%d %d", &A, &B);
	
	Q = A / B;
	R = A % B;
	
	printf("Quotient Q = %d\n", Q);
	printf("Reste R = %d\n", R);
	
	return 0;
	}

►**Lien du programme**

https://onlinegdb.com/Ndx2q3FsJ

Exercice N°03 :

Écrire un algorithme, puis sa traduction en langage C, qui lit la masse d'un objet en kilogrammes et sa vitesse en m/s, puis calcule et affiche son énergie cinétique en joules.

NB : L'énergie cinétique est calculée par la formule suivante :

$$Ec = \frac{1}{2} mV^2 \ ;$$

Avec :
m : masse de l'objet en kg ;
v : vitesse de l'objet en m/s ;
Ec : énergie cinétique en joules (J).

Données :
Masse de l'objet : m=4 kg
Vitesse de l'objet : V=3 m/s

Corrigé :

Algorithme	**Programme C**

Algorithme EnergieCinetique ;
Variables
m, V, Ec : Réels ;
Début
// Entrées
Écrire("Donner la masse m (en kg) :") ;
Lire(m) ;
Écrire("Donner la vitesse V (en m/s) :") ;
Lire(V) ;
 	
// Traitement
Ec ← 0.5 × m × V × V ;
 	
// Sortie
Écrire("L'énergie cinétique Ec = ", Ec, " J") ;
Fin.
	#include <stdio.h>
	#include <math.h>
	int main()
	{
	float m, V, Ec;
	
	printf("Donner la masse m (en kg) : ");
	scanf("%f", &m);
	
	printf("Donner la vitesse V (en m/s) : ");
	scanf("%f", &V);
	
	Ec = 0.5 * m * V * V; ou Ec = 0.5 * m * pow(V,2) ;
	
	printf("L'energie cinetique Ec = %.2f J\n", Ec);
	
	return 0;
	}

►**Lien du programme** : https://onlinegdb.com/69eT6pvylL