

## TP Initiation à la programmation

### Corrigé de la Série de Tp N°01 : – Les instructions de lecture, écriture et affectation en langage C

#### Corrigé de l'exercice N°01:

1. Listez toutes les variables utilisées puis donnez leur type en langage C ;

Variables utilisées dans l'algorithme	Leur type en langage C
<ul style="list-style-type: none"> <li>➤ d, m, V, Ec : réel ;</li> <li>➤ t : Entier;</li> </ul>	<ul style="list-style-type: none"> <li>➤ float d, m, V, Ec ; <b>float</b> signifie <b>nombre réel à virgule flottante</b>.</li> <li>➤ int t ; <b>int</b> est l'abréviation de <b>integer</b> (entier en anglais).</li> </ul>

2. Écrire les instructions C pour lire les données d'entrées (d, t et m) :

Méthode	Instructions en C	Signification des symboles
a) Une instruction par variable	scanf("%f",&d); scanf("%d",&t); scanf("%f",&m);	%f : lire un nombre réel (float) %d : lire un nombre entier (int) & : adresse de la variable, permet de stocker la valeur saisie dans la variable
b) Une seule instruction pour les trois variables	scanf("%f%d%f",&d,&t,&m);	

3. Écrire les instructions C pour afficher les données de sorties (Ec et V) :

Méthode	Instructions en C	Signification des symboles
a) Une instruction par variable	printf("V = %f\n",V); printf("Ec = %f\n", Ec);	%f : afficher un nombre réel (float) \n : retour à la nouvelle ligne
b) Une seule instruction pour les deux variables	printf("V = %f et Ec = %f\n",V, Ec);	

4. Traduire l'algorithme en programme C

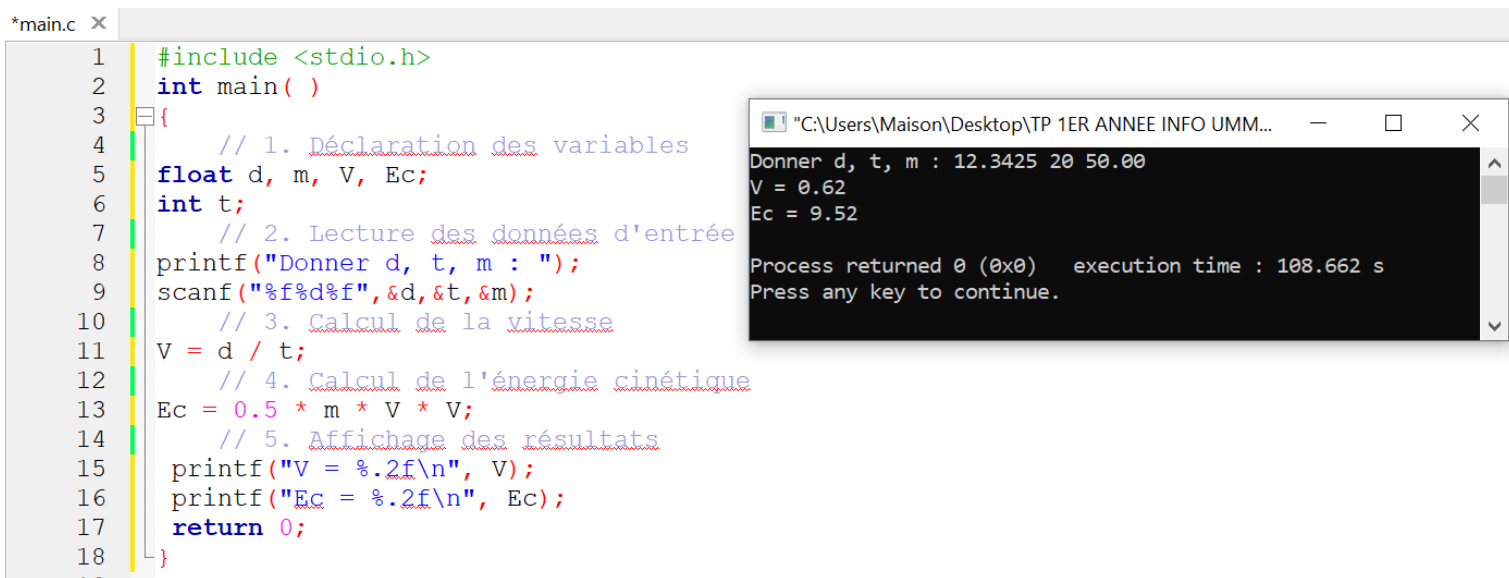
Algorithme	Programme en C
<p><b>Algorithme</b> EnergieCinetique;</p> <p><b>Variables</b></p> <p>d, m, V, Ec : réel ;</p> <p>t : Entier;</p> <p><b>Début</b></p> <p>{-*-*- Entrées-**-*-}</p> <p>Écrire ("Donner d, t, m")</p> <p>Lire (d, t, m)</p> <p>{-*-*- Traitements-**-*-}</p> <p><math>V \leftarrow d / t</math> ;</p> <p><math>Ec \leftarrow 0.5 * m * V * V</math> ;</p> <p>{-*-*- Sorties-**-*-}</p> <p>Ecrire ("V = ", V);</p> <p>Ecrire ("Ec = ", Ec) ;</p> <p><b>Fin.</b></p>	<pre>#include &lt;stdio.h&gt; // Inclusion de la bibliothèque standard pour printf et scanf int main() //int main() représente la fonction principale du programme {     // { → Début du bloc d'instructions ou Début de la fonction principale     // 1. Déclaration des variables     float d, m, V, Ec; // d : distance, m : masse, V : vitesse, Ec : énergie cinétique     int t; // t : temps (entier)     // 2. Lecture des données d'entrée     printf("Donner d, t, m : "); // Affiche un message à l'utilisateur de saisir les valeurs     scanf("%f%d%f",&amp;d,&amp;t,&amp;m); // Lit simultanément d(float),t(int) et m(float) depuis le clavier     // 3. Calcul de la vitesse     V = d / t; // vitesse = distance / temps     // 4. Calcul de l'énergie cinétique     Ec = 0.5 * m * V * V; // Ec = 1/2 * masse * vitesse^2     // 5. Affichage des résultats     printf("V = %f\n", V); // Affiche la vitesse avec 6 décimales par défaut     printf("Ec = %f\n", Ec); // Affiche l'énergie cinétique avec 6 décimales par défaut     return 0; // Fin du programme } // Fin du bloc d'instructions ou Fin de la fonction principale</pre>

## 5. Modifier le Prog-C pour que les résultats soient affichés avec deux décimales.

### Programme en C

```
#include <stdio.h> // Inclusion de la bibliothèque standard pour printf et scanf
int main() //int main() représente la fonction principale du programme
{
    // { → Début du bloc d'instructions ou Début de la fonction principale
    // 1. Déclaration des variables
    float d, m, V, Ec; // d : distance, m : masse, V : vitesse, Ec : énergie cinétique
    int t; // t : temps (entier)
    // 2. Lecture des données d'entrée
    printf("Donner d, t, m : "); // Affiche un message à l'utilisateur de saisir les valeurs
    scanf("%f%d%f", &d, &t, &m); // Lit simultanément d(float), t(int) et m(float) depuis le clavier
    // 3. Calcul de la vitesse
    V = d / t; // vitesse = distance / temps
    // 4. Calcul de l'énergie cinétique
    Ec = 0.5 * m * V * V; // Ec = 1/2 * masse * vitesse^2
    // 5. Affichage des résultats
    printf("V = %.2f\n", V); // Affiche la vitesse avec 2 décimales
    printf("Ec = %.2f\n", Ec); // Affiche l'énergie cinétique avec 2 décimales
    return 0; // Fin du programme
} // Fin du bloc d'instructions ou Fin de la fonction principale
```

## 6. Compiler et exécuter le programme pour : $d = 12,3425$ ; $t = 20$ ; $m = 50,00$ .



The image shows a code editor window titled '\*main.c' with the following C code:

```
1 #include <stdio.h>
2 int main()
3 {
4     // 1. Déclaration des variables
5     float d, m, V, Ec;
6     int t;
7     // 2. Lecture des données d'entrée
8     printf("Donner d, t, m : ");
9     scanf("%f%d%f", &d, &t, &m);
10    // 3. Calcul de la vitesse
11    V = d / t;
12    // 4. Calcul de l'énergie cinétique
13    Ec = 0.5 * m * V * V;
14    // 5. Affichage des résultats
15    printf("V = %.2f\n", V);
16    printf("Ec = %.2f\n", Ec);
17    return 0;
18 }
```

Next to the editor is a terminal window titled '"C:\Users\Maison\Desktop\TP 1ER ANNEE INFO UMM...'. It displays the output of the program:

```
Donner d, t, m : 12.3425 20 50.00
V = 0.62
Ec = 9.52

Process returned 0 (0x0)   execution time : 108.662 s
Press any key to continue.
```

7. **Déduire ce que fait le programme** : Ce programme permet de déterminer la vitesse moyenne d'un mobile à partir de la distance parcourue et du temps de déplacement, puis de calculer son énergie cinétique en fonction de sa masse et de la vitesse obtenue.

## Corrigé de l'exercice N°02 :

Écrire un programme en langage C qui Permet de saisir trois nombres entiers **A**, **B** et **C**. Affiche ensuite un message correspondant à l'une des situations suivantes, en respectant cet ordre de priorité : Si les trois nombres sont nuls, afficher : *{Les trois nombres sont nuls}* ; Sinon, si les trois nombres sont égaux, afficher : *{Les trois nombres sont égaux}* ; Sinon, si deux nombres seulement sont égaux, afficher : *{Deux nombres sont égaux}* ; Sinon, si les trois nombres sont différents, afficher : *{Les trois nombres sont différents}*

## Programme en C

```
#include <stdio.h>
int main() {
    int A, B, C;
    /* Saisie des trois entiers */
    printf("Donner les valeurs de A, B et C : ");
    scanf("%d %d %d", &A, &B, &C);

    /* Tests selon l'ordre de priorité */
    if (A == 0 && B == 0 && C == 0) {
        printf("Les trois nombres sont nuls\n");
    }
    else if (A == B && B == C) {
        printf("Les trois nombres sont egaux\n");
    }
    else if (A == B || A == C || B == C) {
        printf("Deux nombres sont egaux\n");
    }
    else {
        printf("Les trois nombres sont differents\n");
    }
    return 0;
}
```

### Compilation et exécution du programmes ( test sur les 4 cas )

```
main.c X
1 #include <stdio.h>
2 int main() {
3     int A, B, C;
4     /* Saisie des trois entiers */
5     printf("Donner les valeurs de A, B et C : ");
6     scanf("%d %d %d", &A, &B, &C);
7
8     /* Tests selon l'ordre de priorité */
9     if (A == 0 && B == 0 && C == 0) {
10        printf("Les trois nombres sont nuls\n");
11    }
12    else if (A == B && B == C) {
13        printf("Les trois nombres sont egaux\n");
14    }
15    else if (A == B || A == C || B == C) {
16        printf("Deux nombres sont egaux\n");
17    }
18    else {
19        printf("Les trois nombres sont differents\n");
20    }
21    return 0;
22 }
23
```

Terminal 1: Donner les valeurs de A, B et C : 3 9 3  
Deux nombres sont egaux  
Process returned 0 (0x0) execution time : 13.491 s  
Press any key to continue.

Terminal 2: Donner les valeurs de A, B et C : 2 3 90  
Les trois nombres sont differents  
Process returned 0 (0x0) execution time : 7.937 s  
Press any key to continue.

Terminal 3: Donner les valeurs de A, B et C : 0 0 0  
Les trois nombres sont nuls  
Process returned 0 (0x0) execution time : 4.364 s

Terminal 4: Donner les valeurs de A, B et C : 5 5 5  
Les trois nombres sont egaux  
Process returned 0 (0x0) execution time : 2.959 s

## Corrigé de l'exercice N°03 :

### Programme en C

```
#include <stdio.h>
int main() {
    float R;    // Résistance
    float I;    // Courant
    float V;    // Tension
    float P;    // Puissance
    float N;    // Courant maximal
    /* Saisie des données */
    printf("Donner la valeur de la resistance R (en ohms) : ");
    scanf("%f", &R);
    printf("Donner la valeur maximale du courant N (N >= 3) : ");
    scanf("%f", &N);
    /* Vérification de la condition sur N */
    if (N < 3) {
        printf("Erreur : N doit etre superieur ou egal a 3.\n");
        return 1;
    }
    /* Calcul et affichage des resultats */
    printf("\nI (A)\tV (V)\tP (W)\n");
    printf("*****\n");
    for (I = 2.0; I <= N; I += 0.2) { // ou bien {for (I = 2.0; I <= N; I = I + 0.2) }
        V = R * I;
        P = V * I; // ou P = R * I * I
        printf("%.1f\t %.2f\t %.2f\n", I, V, P); // \t insère une tabulation, c'est-à-dire un espace d'alignement
    }
    return 0;
}
```

### Compilation et exécution du programme pour $R=1.2k\Omega$ et $I_{max}=N=6A$

```
main.c x
1 #include <stdio.h>
2 int main() {
3     float R;    // Résistance
4     float I;    // Courant
5     float V;    // Tension
6     float P;    // Puissance
7     float N;    // Courant maximal
8     /* Saisie des données */
9     printf("Donner la valeur de la resistance R (en ohms) : ");
10    scanf("%f", &R);
11    printf("Donner la valeur maximale du courant N (N >= 3) : ");
12    scanf("%f", &N);
13    /* Vérification de la condition sur N */
14    if (N < 3) {
15        printf("Erreur : N doit etre superieur ou egal a 3.\n");
16        return 1;
17    }
18    /* Calcul et affichage des resultats */
19    printf("\n*****\n");
20    printf("\nI (A)\t V (V)\t\t P (W)\n");
21    printf("*****\n");
22    for (I = 2.0; I <= N; I += 0.2) {
23        V = R * I;
24        P = V * I; // ou P = R * I * I
25        printf("%.1f\t %.2f\t %.2f\n", I, V, P);
26    }
27    printf("*****\n");
28    return 0;
29 }
30
```

```
"C:\Users\Maison\Desktop\TP 1ER ANNEE IN... - □ ×
Donner la valeur de la resistance R (en ohms) : 1200
Donner la valeur maximale du courant N (N >= 3) : 6
*****
I (A)      V (V)      P (W)
*****
2.0        2400.00    4800.00
2.2        2640.00    5808.00
2.4        2880.00    6912.00
2.6        3120.00    8112.00
2.8        3360.00    9408.00
3.0        3600.00    10800.00
3.2        3840.00    12288.00
3.4        4080.00    13872.00
3.6        4320.00    15552.00
3.8        4560.00    17328.00
4.0        4800.00    19200.00
4.2        5040.00    21168.00
4.4        5280.00    23232.00
4.6        5520.00    25392.00
4.8        5760.00    27648.00
5.0        6000.00    29999.99
5.2        6240.00    32447.99
5.4        6480.00    34991.99
5.6        6720.00    37631.98
5.8        6960.00    40367.98
6.0        7200.00    43199.98
*****
Process returned 0 (0x0)   execution time : 4.593 s
```