

TP Initiation à la programmation

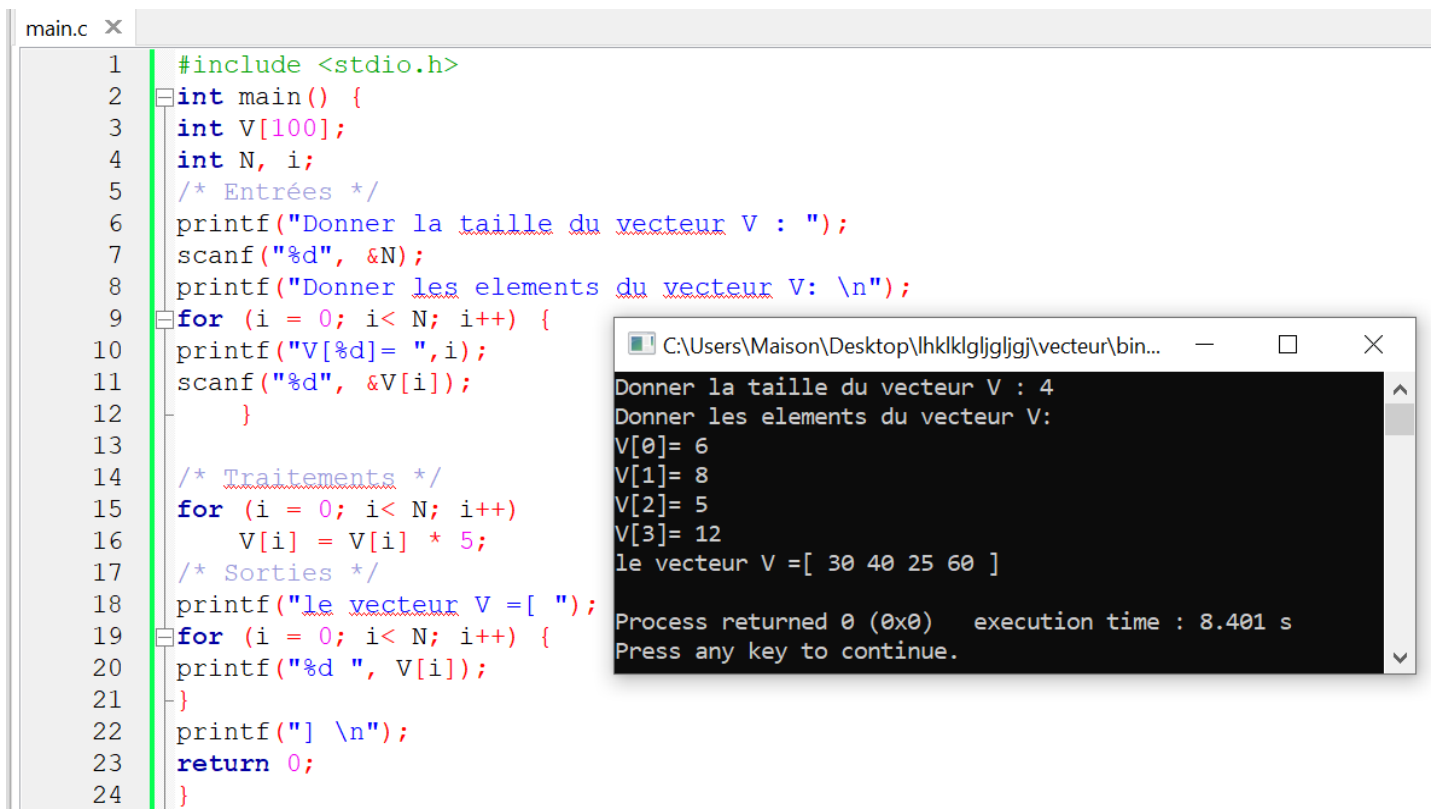
Corrigé de la Série de TP N°1 – Tableaux à une dimension - Vecteurs

Corrigé de l'exercice N°01 :

1. Traduction de l'algorithme :

Algorithme	Langages C
Algorithme Vecteur ; Variables V : Tableau [0..99] d'entier ; N, i : entier ; début // Entrées Écrire("Donner la taille du vecteur V : ") ; Lire(N) ; Écrire("Donner les éléments du vecteur V : ") ; Pour i ← 0 à N-1 faire Écrire("V[i] = ") ; Lire(V[i]) ; Fin-Pour ; // Traitements Pour i ← 0 à N-1 faire V[i] ← V[i] * 5 ; Fin-Pour ; // sorties Écrire("le vecteur V =[") ; Pour i ← 0 à N-1 faire Écrire(V[i]) ; Fin-Pour ; Écrire("]") ; Fin.	<pre>#include <stdio.h> int main() { int V[100]; int N, i; /* Entrées */ printf("Donner la taille du vecteur V : "); scanf("%d", &N); printf("Donner les elements du vecteur V: \n"); for (i = 0; i< N; i++) { printf("V[%d]= ",i); scanf("%d", &V[i]); } /* Traitements */ for (i = 0; i< N; i++) V[i] = V[i] * 5; /* Sorties */ printf("le vecteur V =["); for (i = 0; i< N; i++) { printf("%d ", V[i]); } printf("] \n"); return 0; }</pre>

2. Compiler et exécuter le programme pour les valeurs suivantes : N = 4 et V = [6 8 5 12]



```
main.c x
1  #include <stdio.h>
2  int main() {
3      int V[100];
4      int N, i;
5      /* Entrées */
6      printf("Donner la taille du vecteur V : ");
7      scanf("%d", &N);
8      printf("Donner les elements du vecteur V: \n");
9      for (i = 0; i< N; i++) {
10         printf("V[%d]= ",i);
11         scanf("%d", &V[i]);
12     }
13
14     /* Traitements */
15     for (i = 0; i< N; i++)
16         V[i] = V[i] * 5;
17     /* Sorties */
18     printf("le vecteur V =[ ");
19     for (i = 0; i< N; i++) {
20         printf("%d ", V[i]);
21     }
22     printf("] \n");
23     return 0;
24 }
```

```
C:\Users\Maison\Desktop\lhklgljgljglj\vecteur\bin...
Donner la taille du vecteur V : 4
Donner les elements du vecteur V:
V[0]= 6
V[1]= 8
V[2]= 5
V[3]= 12
le vecteur V =[ 30 40 25 60 ]

Process returned 0 (0x0)   execution time : 8.401 s
Press any key to continue.
```

3. D  rouler le Prog-C pour les valeurs de N = 4 et V = [6 8 5 12] ;

Instructions	Variables			Affichages
	N	i	V	
printf("Donner la taille du vecteur V : ");	/	/	/	Donner la taille du vecteur V :
scanf("%d", &N);	4	/	/	/
printf("Donner les elements du vecteur V: \n");	4	/	/	Donner les elements du vecteur V :
for (i = 0; i < N; i++) { printf("V[%d]= ", i); scanf("%d", &V[i]); }	4	0 1 2 3	6 8 5 12	V[0]= V[1]= V[2]= V[3]=
for i = 0 V[0] = V[0] * 5 ; V[0] = 6 * 5 ; V[0] = 30 ;	4	0	[30 8 5 12]	
for i = 1 V[1] = V[1] * 5 ; V[1] = 8 * 5 ; V[1] = 40 ;	4	1	[30 40 5 12]	
for i = 2 V[2] = V[2] * 5 ; V[2] = 5 * 5 ; V[2] = 25 ;	4	2	[30 40 25 12]	
for i = 3 V[3] = V[3] * 5 ; V[3] = 12 * 5 ; V[3] = 60 ;	4	3	[30 40 25 60]	
printf("le vecteur V =["); for (i = 0; i < N; i++) { printf("%d ", V[i]); printf("] \n");	4	3	[30 40 25 60]	le vecteur V = [30 40 25 60]

Sans l'instruction printf

```
for (i = 0; i < N; i++) {  
scanf("%d", &V[i]);  
}
```

4

0
1
2
3

6
8
5
12

/

4. D  duire ce que fait cet algorithme: Le programme multiplie les   l  ments du vecteur V par 5.

5. R   crire le programme en rempla  ant la boucle for par la boucle while dans la partie traitements.

Algorithme	Programme en C
Algorithme VecteurQuest5 ; Variables V : tableau[0 ..99] d'entiers ; N, i : entier ; D��but // Entr��e de la taille du vecteur ��crire("Donner la taille du vecteur V : ") ; Lire(N) ; // Saisie des ��l��ments du vecteur ��crire("Donner les ��l��ments du vecteur V : ") ; Pour i �� 0 �� N-1 Faire ��crire("V[i] = ") ; Lire(V[i]) ; Fin-Pour ; // Traitement i �� 0 ; Tantque i < N Faire V[i] �� V[i] * 5 ; i �� i + 1 ; Fin-Tantque ; // Affichage du vecteur ��crire("Le vecteur V = [") ; Pour i �� 0 �� N-1 Faire ��crire(" V[i] ") ; Fin-Pour ; ��crire("] ") ; Fin.	#include <stdio.h> int main() { int V[100]; int N, i; /* Entr��es */ printf("Donner la taille du vecteur V : "); scanf("%d", &N); printf("Donner les elements du vecteur V: \n"); for (i = 0; i < N; i++) { printf("V[%d]= ", i); scanf("%d", &V[i]); } /* Traitements avec boucle while */ i = 0; // initialisation du compteur while (i < N) { V[i] = V[i] * 5; i++; // incr��ment du compteur } /* Sorties */ printf("Le vecteur V =["); for (i = 0; i < N; i++) { printf("%d ", V[i]); } printf("] \n"); return 0; }

Corrigé de l'exercice N°02 :

- Écrire un algorithme/programme C qui permet de calculer la somme et la moyenne des éléments d'un vecteur V réels.

Algorithme	Programme en C
Algorithme SommeMoyenneVecteur ; Variables V : tableau[100] de réels ; N, i : entier ; somme, moyenne : réel ; Début <i>// Saisie de la taille du vecteur</i> Écrire("Donner la taille du vecteur V : ") ; Lire(N) ; <i>// Saisie des éléments du vecteur</i> Pour i ← 0 à N-1 Faire Écrire("V[" , i , "] = ") ; Lire(V[i]) ; FinPour ; <i>// Calcul de la somme</i> somme ← 0 ; Pour i ← 0 à N-1 Faire somme ← somme + V[i] ; FinPour ; <i>// Calcul de la moyenne</i> moyenne ← somme / N ; <i>// Affichage des résultats</i> Écrire("La somme des éléments du vecteur V = ", somme) ; Écrire("La moyenne des éléments du vecteur V = ", moyenne) Fin.	<pre>#include <stdio.h> int main() { int N, i; float V[100], somme = 0, moyenne; // Saisie de la taille du vecteur printf("Donner la taille du vecteur V : "); scanf("%d", &N); // Saisie des éléments du vecteur printf("Donner les elements du vecteur V :\n"); for(i = 0; i < N; i++) { printf("V[%d] = ", i); scanf("%f", &V[i]); } // Calcul de la somme for(i = 0; i < N; i++) { somme += V[i]; //somme = somme +V[i]; } // Calcul de la moyenne moyenne = somme / N; // Affichage des résultats printf("La somme des éléments du vecteur V = %.2f\n", somme); printf("La moyenne des éléments du vecteur V = %.2f\n", moyenne); return 0; }</pre>

Compilation et exécution pour N=5 et V= [1 3 5 7 9]

```
main.c x
1  #include <stdio.h>
2  int main() {
3      int N, i;
4      float V[100], somme = 0, moyenne;
5      // Saisie de la taille du vecteur
6      printf("Donner la taille du vecteur V : ");
7      scanf("%d", &N);
8      // Saisie des éléments du vecteur
9      printf("Donner les elements du vecteur V :\n");
10     for(i = 0; i < N; i++) {
11         printf("V[%d] = ", i);
12         scanf("%f", &V[i]);
13     }
14     // Calcul de la somme
15     for(i = 0; i < N; i++) {
16         somme += V[i]; //somme = somme +V[i];
17     }
18     // Calcul de la moyenne
19     moyenne = somme / N;
20
21     // Affichage des résultats
22     printf("La somme des elements du vecteur V=%.2f\n", somme);
23     printf("La moyenne des elements du vecteur V=%.2f\n", moyenne);
24     return 0;
25 }
```

```
C:\Users\Maison\Desktop\lhklgljglgj\vecteur\...
Donner la taille du vecteur V : 5
Donner les elements du vecteur V :
V[0] = 1
V[1] = 3
V[2] = 5
V[3] = 7
V[4] = 9
La somme des elements du vecteur V=25.00
La moyenne des elements du vecteur V=5.00
Process returned 0 (0x0)   execution time : 10.274 s
Press any key to continue.
```

- améliorer le programme afin de calculer également la **variance** σ^2 et l'**écart-type** σ du vecteur, et d'afficher tous ces résultats. **Rappel** : $\sigma^2 = \frac{1}{n} \sum_{i=0}^{n-1} (V_i - \text{moyenne})^2$;

Langage C

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

int main() {
    int N, i;
    float V[100], somme = 0, moyenne, variance = 0, ecartType;
    // Saisie de la taille du vecteur
    printf("Donner la taille du vecteur V : ");
    scanf("%d", &N);
    // Saisie des éléments du vecteur
    printf("Donner les elements du vecteur V :\n");
    for(i = 0; i < N; i++) {
        printf("V[%d] = ", i);
        scanf("%f", &V[i]);
    }
    // Calcul de la somme
    for(i = 0; i < N; i++) {
        somme += V[i];
    }
    // Calcul de la moyenne
    moyenne = somme / N;
    // Calcul de la variance
    for(i = 0; i < N; i++) {
        variance += pow(V[i] - moyenne, 2); // variance = variance + pow(V[i] - moyenne, 2);
    }
    variance = variance / N;
    // Calcul de l'écart-type
    ecartType = sqrt(variance);
    // Affichage des résultats
    printf("La somme des éléments du vecteur V = %.2f\n", somme);
    printf("La moyenne des éléments du vecteur V = %.2f\n", moyenne);
    printf("La variance du vecteur V = %.2f\n", variance);
    printf("L'ecart-type du vecteur V = %.2f\n", ecartType);
    return 0;
}
```

Compilation et exécution pour N=5 et V= [1 3 5 7 9]

```
main.c x
1 #include <stdio.h>
2 #include <math.h>
3 int main() {
4     int N, i;
5     float V[100], somme = 0, moyenne, variance = 0, ecartType;
6     // Saisie de la taille du vecteur
7     printf("Donner la taille du vecteur V : ");
8     scanf("%d", &N);
9     // Saisie des éléments du vecteur
10    printf("Donner les elements du vecteur V :\n");
11    for(i = 0; i < N; i++) {
12        printf("V[%d] = ", i);
13        scanf("%f", &V[i]);
14    }
15    // Calcul de la somme
16    for(i = 0; i < N; i++) {
17        somme += V[i];
18    }
19    // Calcul de la moyenne
20    moyenne = somme / N;
21
22    // Calcul de la variance
23    for(i = 0; i < N; i++) {
24        variance += pow(V[i] - moyenne, 2); //variance=variance+pow(V[i]-moyenne, 2);
25    }
26    variance = variance / N;
27    // Calcul de l'écart-type
28    ecartType = sqrt(variance);
29    // Affichage des résultats
30    printf("La somme des elements du vecteur V = %.2f\n", somme);
31    printf("La moyenne des elements du vecteur V = %.2f\n", moyenne);
32    printf("La variance du vecteur V = %.2f\n", variance);
33    printf("L'ecart-type du vecteur V = %.2f\n", ecartType);
34    return 0;
35 }
```

```
C:\Users\Maison\Desktop\lhklgljlgj\vecteur...
Donner la taille du vecteur V : 5
Donner les elements du vecteur V :
V[0] = 1
V[1] = 3
V[2] = 5
V[3] = 7
V[4] = 9
La somme des elements du vecteur V = 249.00
La moyenne des elements du vecteur V = 49.80
La variance du vecteur V = 2015.04
L'ecart-type du vecteur V = 44.89

Process returned 0 (0x0)   execution time : 7.064 s
Press any key to continue.
```

Corrigé de l'exercice N°03 :

On saisit n positions $y[i]$ mesurées à intervalles de temps constants dt . On calcule ensuite le vecteur vitesse V entre deux mesures successives avec la formule : $v_i = (y_{i+1} - y_i) / dt$.

On a :

- n positions mesurées à intervalles de temps constants dt .
- Ces positions sont stockées dans un **vecteur** y : $y[0], y[1], \dots, y[n-1]$.
- On souhaite calculer la **vitesse moyenne entre deux positions successives**, stockée dans un vecteur V .

Formule de la vitesse

La vitesse entre deux mesures successives est donnée par :

$$v_i = (y_{i+1} - y_i) / dt .$$

- $v_0 \rightarrow$ vitesse entre $y[0]$ et $y[1]$
- $v_1 \rightarrow$ vitesse entre $y[1]$ et $y[2]$
- ...
- Il y a donc **$n-1$ vitesses** pour n positions.

Programme en C

```
#include <stdio.h>
int main( ) {
    int n, i;
    float dt;
    float y[100]; // tableau pour les positions
    float V[99];  // tableau pour les vitesses
    // Saisie du nombre de positions
    printf("Donner le nombre de positions n : ");
    scanf("%d", &n);
    // Vérification de la taille
    if(n < 2 || n > 100) {
        printf("Erreur : n doit être compris entre 2 et 100.\n");
        return 1;
    }
    // Saisie de l'intervalle de temps
    printf("Donner l'intervalle de temps dt : ");
    scanf("%f", &dt);

    // Saisie des positions
    printf("Saisir les positions y[i] :\n");
    for(i = 0; i < n; i++) {
        printf("y[%d] = ", i);
        scanf("%f", &y[i]);
    }
    // Calcul des vitesses
    for(i = 0; i < n - 1; i++) {
        V[i] = (y[i+1] - y[i]) / dt;
    }
    // Affichage des vitesses
    printf("\nVecteur vitesse V :\n");
    for(i = 0; i < n - 1; i++) {
        printf("v[%d] = %.2f\n", i, V[i]);
    }
    return 0;
}
```

```

1 #include <stdio.h>
2 int main() {
3     int n, i;
4     float dt;
5     float y[100]; // tableau pour les positions
6     float V[99]; // tableau pour les vitesses
7     // Saisie du nombre de positions
8     printf("Donner le nombre de positions n : ");
9     scanf("%d", &n);
10    // Vérification de la taille
11    if(n < 2 || n > 100) {
12        printf("Erreur : n doit être compris entre 2 et 100.\n");
13        return 1;
14    }
15    // Saisie de l'intervalle de temps
16    printf("Donner l'intervalle de temps dt : ");
17    scanf("%f", &dt);
18
19    // Saisie des positions
20    printf("Saisir les positions y[i] :\n");
21    for(i = 0; i < n; i++) {
22        printf("y[%d] = ", i);
23        scanf("%f", &y[i]);
24    }
25    // Calcul des vitesses
26    for(i = 0; i < n - 1; i++) {
27        V[i] = (y[i+1] - y[i]) / dt;
28    }
29    // Affichage des vitesses
30    printf("\nVecteur vitesse V :\n");
31    for(i = 0; i < n - 1; i++) {
32        printf("v[%d] = %.2f\n", i, V[i]);
33    }
34    return 0;
35 }

```

```

C:\Users\Maison\Desktop\lhklgljgljg\vecteur\b...
Donner le nombre de positions n : 6
Donner l'intervalle de temps dt : 2
Saisir les positions y[i] :
y[0] = 1
y[1] = 4
y[2] = 9
y[3] = 19
y[4] = 38
y[5] = 72

Vecteur vitesse V :
v[0] = 1.50
v[1] = 2.50
v[2] = 5.00
v[3] = 9.50
v[4] = 17.00

Process returned 0 (0x0)   execution time : 92.872 s
Press any key to continue.

```

Corrigé de l'exercice N°04 :

Programme en C

```

#include <stdio.h>
int main() {
    int n, i;
    float A[100], B[100], Psca = 0;
    // Saisie de la dimension
    printf("Donner la dimension des vecteurs n : ");
    scanf("%d", &n);
    // Vérification
    if(n <= 0 || n > 100) {
        printf("Erreur : n doit être compris entre 1 et 100.\n");
        return 1;
    }
    // Saisie des éléments du vecteur A
    printf("Saisir les éléments du vecteur A :\n");
    for(i = 0; i < n; i++) {
        printf("A[%d] = ", i);
        scanf("%f", &A[i]);
    }
    // Saisie des éléments du vecteur B
    printf("Saisir les éléments du vecteur B :\n");
    for(i = 0; i < n; i++) {
        printf("B[%d] = ", i);
        scanf("%f", &B[i]);
    }
    // Calcul du produit scalaire
    for(i = 0; i < n; i++) {
        Psca += A[i] * B[i];
    }
    // Affichage du résultat
    printf("\nLe produit scalaire des deux vecteurs est : %.2f\n", Psca);
    return 0;
}

```

```

1  #include <stdio.h>
2  int main() {
3      int n, i;
4      float A[100], B[100], Psca = 0;
5      // Saisie de la dimension
6      printf("Donner la dimension des vecteurs n : ");
7      scanf("%d", &n);
8      // Vérification
9      if(n <= 0 || n > 100) {
10         printf("Erreur : n doit être compris entre 1 et 100.\n");
11         return 1;
12     }
13     // Saisie des éléments du vecteur A
14     printf("Saisir les éléments du vecteur A :\n");
15     for(i = 0; i < n; i++) {
16         printf("A[%d] = ", i);
17         scanf("%f", &A[i]);
18     }
19     // Saisie des éléments du vecteur B
20     printf("Saisir les éléments du vecteur B :\n");
21     for(i = 0; i < n; i++) {
22         printf("B[%d] = ", i);
23         scanf("%f", &B[i]);
24     }
25     // Calcul du produit scalaire
26     for(i = 0; i < n; i++) {
27         Psca += A[i] * B[i];
28     }
29     // Affichage du résultat
30     printf("\nLe produit scalaire des deux vecteurs est : %.2f", Psca);
31     return 0;

```

```

C:\Users\Maison\Desktop\lhklklgljgljglj\vecteur\...
Donner la dimension des vecteurs n : 5
Saisir les éléments du vecteur A :
A[0] = 1
A[1] = 3
A[2] = 2
A[3] = 5
A[4] = 10
Saisir les éléments du vecteur B :
B[0] = 1
B[1] = 2
B[2] = 3
B[3] = 4
B[4] = 5

Le produit scalaire des deux vecteurs est : 83.00

Process returned 0 (0x0)   execution time : 30.863 s
Press any key to continue.

```

Vérification par un calcul numérique du produit scalaire pour : A= [1 3 2 5 10], B=[1 2 3 4 5]

Le produit scalaire A·B est :

$$Psca = \sum_{i=0}^{n-1} A[i] * B[i]$$

Calcul :

- 1*1=1
- 3*2=6
- 2*3=6
- 5*4=20
- 10*5=50

Addition : **Psca = 1 + 6 + 6 + 20 + 50 = 83**

Corrigé de l'exercice N°05 :

Variables : T : Vecteur ; n : taille du vecteur ; nPos : nombres positifs ; nNeg : nombres Négatifs ; nNul : nombres Nuls

Programme en C

```
#include <stdio.h>
int main() {
    int n, i;
    int nPos = 0, nNeg = 0, nNul = 0; // Compteurs pour positifs, negatifs et nuls
    float T[100];
    // Saisie de la taille du vecteur
    printf("Donner le nombre n d'elements du vecteur T : ");
    scanf("%d", &n);
    if(n <= 0 || n > 100) {
        printf("Erreur : la taille doit etre comprise entre 1 et 100.\n");
        return 1;
    }
    // Saisie des éléments du vecteur T
    printf("Saisir les elements du vecteur T :\n");
    for(i = 0; i < n; i++) {
        printf("T[%d] = ", i);
        scanf("%f", &T[i]);
    }
    // Comptage des éléments positifs, négatifs et nuls
    for(i = 0; i < n; i++) {
        if(T[i] > 0)
            nPos++; // nPos = nPos+1 ;
        else if(T[i] < 0)
            nNeg++; // nNeg = nNeg+1 ;
        else
            nNul++; // nNul = nNul+1 ;
    }
    // Affichage des résultats
    printf("\nNombre d'elements positifs : %d\n", nPos);
    printf("Nombre d'elements negatifs : %d\n", nNeg);
    printf("Nombre d'elements nuls : %d\n", nNul);
    return 0;}
```

Compilation et exécution pour n=6 et T= [1 0 -5 0 9 20]

```
main.c x
1 #include <stdio.h>
2 int main() {
3     int n, i;
4     int nPos = 0, nNeg = 0, nNul = 0; // Compteurs pour positifs, negatifs et nuls
5     float T[100];
6     // Saisie de la taille du vecteur
7     printf("Donner le nombre n d'elements du vecteur T : ");
8     scanf("%d", &n);
9     if(n <= 0 || n > 100) {
10         printf("Erreur : la taille doit etre comprise entre 1 et 100.\n");
11         return 1;
12     }
13     // Saisie des éléments du vecteur T
14     printf("Saisir les elements du vecteur T :\n");
15     for(i = 0; i < n; i++) {
16         printf("T[%d] = ", i);
17         scanf("%f", &T[i]);
18     }
19     // Comptage des éléments positifs, négatifs et nuls
20     for(i = 0; i < n; i++) {
21         if(T[i] > 0)
22             nPos++; // nPos = nPos+1 ;
23         else if(T[i] < 0)
24             nNeg++; // nNeg = nNeg+1 ;
25         else
26             nNul++; // nNul = nNul+1 ;
27     }
28     // Affichage des résultats
29     printf("\nNombre d'elements positifs : %d\n", nPos);
30     printf("Nombre d'elements negatifs : %d\n", nNeg);
31     printf("Nombre d'elements nuls : %d\n", nNul);
32     return 0;
33 }
```

```
C:\Users\Maison\Desktop\lhklklgljlgj\vecteur\...
Donner le nombre n d'elements du vecteur T : 6
Saisir les elements du vecteur T :
T[0] = 1
T[1] = 0
T[2] = -5
T[3] = 0
T[4] = 9
T[5] = 20

Nombre d'elements positifs : 3
Nombre d'elements negatifs : 1
Nombre d'elements nuls : 2

Process returned 0 (0x0)   execution time : 31.904 s
Press any key to continue.
```