

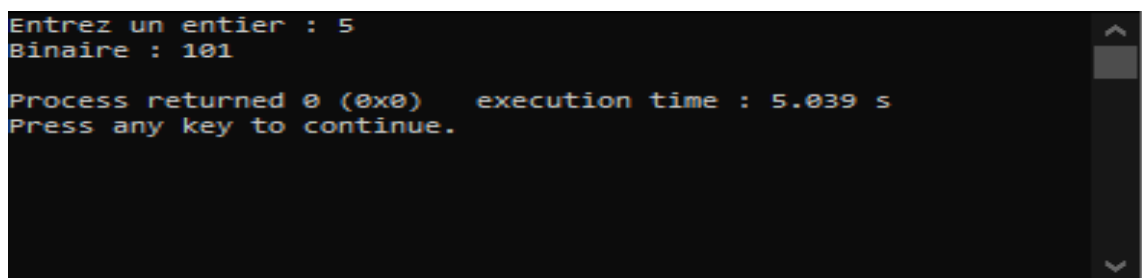
# Travail à rendre :

**Exercice 1.** *Écrire une fonction qui prend en paramètre un entier et renvoie son code binaire ?*

➔ **Solution :**

```
1. #include <stdio.h>
2.
3. void afficherBinaire(int n) {
4.     int binaire[32];
5.     int i = 0;
6.     if (n == 0) {
7.         printf("0\n");
8.         return;
9.     }
10.    while (n > 0) {
11.        binaire[i] = n % 2;
12.        n = n / 2;
13.        i++;
14.    }
15.    printf("Binaire : ");
16.    for (int j = i - 1; j >= 0; j--) {
17.        printf("%d", binaire[j]);
18.    }
19.    printf("\n");
20. }
21. int main() {
22.     int nombre;
23.     printf("Entrez un entier : ");
24.     scanf("%d", &nombre);
25.     afficherBinaire(nombre);
26.     return 0;
27. }
28.
```

➔ **Sortie de Program :**



```
Entrez un entier : 5
Binaire : 101

Process returned 0 (0x0)   execution time : 5.039 s
Press any key to continue.
```

**Exercice 2.** *Écrire une fonction qui prend en paramètre un entier et alloue un tableau de N éléments initialisés à zéro?*

➔ **Solution :**

```

1. #include <stdio.h>
2. #include <stdlib.h>
3. int* creerTableauZero(int n) {
4.     int *tab = calloc(n, sizeof *tab);
5.     if (!tab) {
6.         printf("Erreur : allocation de mémoire impossible.\n");
7.         return NULL;
8.     }
9.     return tab;
10.}
11.
12.int main(void) {
13.    int N;
14.    printf("Entrez un entier N (nombre d'éléments) : ");
15.    if (scanf("%d", &N) != 1 || N < 1) {
16.        printf("Valeur invalide.\n");
17.        return 1;
18.    }
19.    int *tableau = creerTableauZero(N);
20.    if (!tableau) {
21.        return 1;
22.    }
23.    printf("Contenu du tableau initialisé à 0 :\n");
24.    for (int i = 0; i < N; i++) {
25.        printf("tableau[%d] = %d\n", i, tableau[i]);
26.    }
27.
28.    free(tableau);
29.    return 0;
30.}
31.

```

➔ **Sortie de Program :**

```

Entrez un entier : 6
L'adresse du tableau est : 0000000000CB6A70
c[0] = 0
c[1] = 13
c[2] = 26
c[3] = 39
c[4] = 52
c[5] = 65

```

**Exercice 3** . Écrire une fonction qui prend en paramètre un tableau d'entiers, son nombre d'éléments  $n$  et un entier  $m$ , et qui retourne le nombre d'occurrence de cet élément  $m$  dans le tableau, 0 s'il n'existe pas ?

➔ **Solution :**

```

1. #include <stdio.h>
2.
3. int compterOccurrences(int tableau[], int n, int m) {
4.     int compteur = 0;
5.
6.     for (int i = 0; i < n; i++) {
7.         if (tableau[i] == m) {
8.             compteur++;
9.         }
10.    }
11.    return compteur;
12.}
13.int main() {
14.    int n, m;
15.    // Lecture de la taille du tableau et de l'élément à chercher
16.    printf("Entrez le nombre d'éléments du tableau : ");
17.    scanf("%d", &n);
18.    int tableau[n];
19.    printf("Entrez les éléments du tableau :\n");
20.    for (int i = 0; i < n; i++) {
21.        scanf("%d", &tableau[i]);
22.    }
23.    printf("Entrez l'élément m à rechercher : ");
24.    scanf("%d", &m);
25.    int occurrences = compterOccurrences(tableau, n, m);
26.
27.    printf("L'élément %d apparaît %d fois dans le tableau.\n", m, occurrences);
28.    return 0;
29.}
30.

```

➔ **Sortie de Program :**

```

Entrez le nombre a chercher : 9
Le nombre 9 apparait 1 fois dans le tableau.

Process returned 0 (0x0)   execution time : 4.140 s
Press any key to continue.

```

**Exercice 4** . Écrire une fonction qui prend en paramètre une chaîne de caractères et un caractère donné, et renvoie le nombre d'occurrences correspondant à ce caractère dans la chaîne ?

➔ **Solution :**

```

1. #include <stdio.h>
2. #include <string.h>
3.
4. int compterOccurrencesChar( char *str, char c) {
5.     int count = 0;
6.     for (int i = 0; str[i] != '\0'; i++) {
7.         if (str[i] == c) {
8.             count++;
9.         }
10.    }
11.    return count;
12. }
13.
14. int main() {
15.     char chaine[100], caractere;
16.     printf("Entrez une phrase : ");
17.     scanf(" %[^\n]", chaine);
18.
19.     printf("Entrez le caractere a rechercher : ");
20.     scanf(" %c", &caractere);
21.
22.     printf("Occurrences du caractere '%c' : %d\n", caractere,
23.           compterOccurrencesChar(chaine, caractere));
24.     return 0;
25. }

```

➔ **Sortie de Program :**

```

Entrez une phrase : La voiture est rouge
Entrez le caractere a rechercher : o
Occurrences du caractere 'o' : 2

Process returned 0 (0x0)   execution time : 39.144 s
Press any key to continue.

```

## Exercice 5 :

Écrire un programme qui crée un pointeur des pointeurs de type int.

→ Solution :

```

1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. int main() {
5.     int N, i, j;
6.     printf("Entrez le nombre de lignes : ");
7.     scanf("%d", &N);
8.     int **tableau = malloc(N * sizeof(int *));
9.     int *colonnes = (int *)malloc(N * sizeof(int));
10.    // 2. Saisie du nombre de colonnes pour chaque ligne
11.    for (i = 0; i < N; i++) {
12.        printf("Entrez le nombre de colonnes pour la ligne %d : ", i + 1);
13.        scanf("%d", &colonnes[i]);
14.        // Allocation mémoire pour chaque ligne
15.        tableau[i] = (int *)malloc(colonnes[i] * sizeof(int));
16.        // Remplir les éléments
17.        for (j = 0; j < colonnes[i]; j++) {
18.            printf("Entrez l'element [%d][%d] : ", i, j);
19.            scanf("%d", &tableau[i][j]);
20.        }
21.    }
22.    // 3. Affichage du tableau
23.    printf("\nContenu du tableau :\n");
24.    for (i = 0; i < N; i++) {
25.        for (j = 0; j < colonnes[i]; j++) {
26.            printf("%d\t", tableau[i][j]);
27.        }
28.        printf("\n");
29.    }
30.    // 4. Libération de la mémoire
31.    for (i = 0; i < N; i++) {
32.        free(tableau[i]);
33.    }
34.    free(tableau);
35.    free(colonnes);
36.    return 0;
37. }
```

**➔ Sortie de Program :**

```
Entrez le nombre de lignes : 3
Entrez le nombre de colonnes pour la ligne 1 : 3
Entrez l'element [0][0] : 25
Entrez l'element [0][1] : 26
Entrez l'element [0][2] : 27
Entrez le nombre de colonnes pour la ligne 2 : 3
Entrez l'element [1][0] : 16
Entrez l'element [1][1] : 17
Entrez l'element [1][2] : 18
Entrez le nombre de colonnes pour la ligne 3 : 2
Entrez l'element [2][0] : 11
Entrez l'element [2][1] : 12
```

Contenu du tableau :

25	26	27
16	17	18
11	12	

```
Process returned 0 (0x0)   execution time : 32.641 s
Press any key to continue.
```

## Exercice 6 :

Écrire un programme qui crée un pointeur de pointeurs de type char.

### ➔ Solution :

```

1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. int main() {
5.     int lignes, i;
6.     printf("Entrez le nombre de phrases : ");
7.     scanf("%d", &lignes);
8.
9.     char **phrases = (char **)malloc(lignes * sizeof(char *));
10.    if (phrases == NULL) {
11.        printf("Erreur d'allocation mémoire.\n");
12.        return 1;
13.    }
14.
15.    for (i = 0; i < lignes; i++) {
16.        int taille;
17.        printf("Entrez le nombre de caractères pour la phrase %d : ", i + 1);
18.        scanf("%d", &taille);
19.
20.        phrases[i] = (char *)malloc((taille + 1) * sizeof(char));
21.        if (phrases[i] == NULL) {
22.            printf("Erreur d'allocation mémoire.\n");
23.            return 1;
24.        }
25.        printf("Entrez la phrase %d : ", i + 1);
26.        scanf(" %[^\n]", phrases[i]);
27.    }
28.    printf("\nListe des phrases :\n");
29.    for (i = 0; i < lignes; i++) {
30.        printf("Phrase %d : %s\n", i + 1, phrases[i]);
31.    }
32.
33.    // Libération de la mémoire
34.    for (i = 0; i < lignes; i++) {
35.        free(phrases[i]);
36.    }
37.    free(phrases); // Libérer le tableau principal
38.    return 0;
39. }
40.

```

## ➔ Sortie de Program :

```
Entrez le nombre de phrases : 3
Entrez le nombre de caractPres pour la phrase 1 : 7
Entrez la phrase 1 : Voiture
Entrez le nombre de caractPres pour la phrase 2 : 5
Entrez la phrase 2 : human
Entrez le nombre de caractPres pour la phrase 3 : 4
Entrez la phrase 3 : Code

Liste des phrases :
Phrase 1 : Voiture
Phrase 2 : human
Phrase 3 : Code

Process returned 0 (0x0)   execution time : 73.159 s
Press any key to continue.
```



## **Exercice 7** . Écrire deux programmes qui gèrent le classement des équipes d'un championnat donné en utilisant les tableaux ou les pointeurs.:

### ➔ Solution Program 1:

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

#define NOMBRE_EQUIPES 7

void saisirEquipes(char noms[][50], int scores[], int n) {
    for (int i = 0; i < n; i++) {
        printf("Entrez le nom de l'Equipe %d : ", i + 1);
        scanf(" %[^\\n]", noms[i]);
        printf("Entrez le score de l'Equipe %d : ", i + 1);
        scanf("%d", &scores[i]);
    }
}

void trierEquipes(char noms[][50], int scores[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (scores[i] < scores[j]) {
                int tempScore = scores[i];
                scores[i] = scores[j];
                scores[j] = tempScore;
                char tempNom[50];
                strcpy(tempNom, noms[i]);
                strcpy(noms[i], noms[j]);
                strcpy(noms[j], tempNom);
            }
        }
    }
}

void afficherClassement(char noms[][50], int scores[], int n) {
    printf("\\nClassement des Equipes :\\n");
    for (int i = 0; i < n; i++) {
        printf("%d. %s : %d points\\n", i + 1, noms[i], scores[i]);
    }
}

int main() {
    char noms[NOMBRE_EQUIPES][50];
    int scores[NOMBRE_EQUIPES];
    printf("Saisissez les informations des Equipes (7 Equipes) :\\n");
    saisirEquipes(noms, scores, NOMBRE_EQUIPES);
    trierEquipes(noms, scores, NOMBRE_EQUIPES);
    afficherClassement(noms, scores, NOMBRE_EQUIPES);
    printf("\\nClassement des cinq premieres Equipes :\\n");
    afficherClassement(noms, scores, 5);
    return 0;
}
```

**➔ Sortie de Program 1 :**

Saisissez les informations des Equipes (7 Equipes) :

Entrez le nom de l'Equipe 1 : Equipe1

Entrez le score de l'Equipe 1 : 10

Entrez le nom de l'Equipe 2 : Equipe2

Entrez le score de l'Equipe 2 : 20

Entrez le nom de l'Equipe 3 : Equipe3

Entrez le score de l'Equipe 3 : 30

Entrez le nom de l'Equipe 4 : Equipe4

Entrez le score de l'Equipe 4 : 40

Entrez le nom de l'Equipe 5 : Equipe5

Entrez le score de l'Equipe 5 : 50

Entrez le nom de l'Equipe 6 : Equipe6

Entrez le score de l'Equipe 6 : 60

Entrez le nom de l'Equipe 7 : Equipe7

Entrez le score de l'Equipe 7 : 70

Classement des Equipes :

1. Equipe7 : 70 points

2. Equipe6 : 60 points

3. Equipe5 : 50 points

4. Equipe4 : 40 points

5. Equipe3 : 30 points

6. Equipe2 : 20 points

7. Equipe1 : 10 points

Classement des cinq premieres Equipes :

Classement des Equipes :

1. Equipe7 : 70 points

2. Equipe6 : 60 points

3. Equipe5 : 50 points

4. Equipe4 : 40 points

5. Equipe3 : 30 points

## ➔ Solution Program 2 :

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
    char nom[50];
    int score;
} Equipe;

void saisirEquipes(Equipe *equipes, int n) {
    for (int i = 0; i < n; i++) {
        printf("Entrez le nom de l'Equipe %d : ", i + 1);
        scanf("%[^\n]", equipes[i].nom);
        printf("Entrez le score de l'Equipe %d : ", i + 1);
        scanf("%d", &equipes[i].score);
    }
}

void trierEquipes(Equipe *equipes, int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (equipes[i].score < equipes[j].score) {
                Equipe temp = equipes[i];
                equipes[i] = equipes[j];
                equipes[j] = temp;
            }
        }
    }
}

void afficherClassement(Equipe *equipes, int n) {
    printf("\nClassement des Equipes :\n");
    for (int i = 0; i < n; i++) {
        printf("%d. %s : %d points\n", i + 1, equipes[i].nom, equipes[i].score);
    }
}

int main() {
    int n = 7; // Nombre d'équipes
    Equipe *equipes = (Equipe *)malloc(n * sizeof(Equipe)); // Allocation dynamique
    if (equipes == NULL) {
        printf("Erreur d'allocation memoire.\n");
        return 1;
    }
    printf("Saisissez les informations des Equipes (7 Equipes) :\n");
    saisirEquipes(equipes, n);
    trierEquipes(equipes, n);
    afficherClassement(equipes, n);
    // Exemple pour les cinq premières équipes
    printf("\nClassement des cinq premieres equipes :\n");
    afficherClassement(equipes, 5);
    free(equipes);
    return 0;
}

```

## ➔ Sortie de Program 2 :

Saisissez les informations des Equipes (7 Equipes) :

```
Entrez le nom de l'Equipe 1 : EQ1
Entrez le score de l'Equipe 1 : 223
Entrez le nom de l'Equipe 2 : EQ2
Entrez le score de l'Equipe 2 : 334
Entrez le nom de l'Equipe 3 : EQ3
Entrez le score de l'Equipe 3 : 4335
Entrez le nom de l'Equipe 4 : EQ4
Entrez le score de l'Equipe 4 : 565
Entrez le nom de l'Equipe 5 : EQ5
Entrez le score de l'Equipe 5 : 766
Entrez le nom de l'Equipe 6 : EQ6
Entrez le score de l'Equipe 6 : 978
Entrez le nom de l'Equipe 7 : EQ7
Entrez le score de l'Equipe 7 : 444
```

Classement des Equipes :

```
1. EQ3 : 4335 points
2. EQ6 : 978 points
3. EQ5 : 766 points
4. EQ4 : 565 points
5. EQ7 : 444 points
6. EQ2 : 334 points
7. EQ1 : 223 points
```

Classement des cinq premieres equipes :

Classement des Equipes :

```
1. EQ3 : 4335 points
2. EQ6 : 978 points
3. EQ5 : 766 points
4. EQ4 : 565 points
5. EQ7 : 444 points
```

## Exercice 8 . a)

### → Solution :

```
#include <stdio.h>
#define MAX 10 // Taille maximale du tableau

// Fonction qui calcule  $f(x) = x^2 + 4x + 1$  pour un tableau
// donné
void calculF(int x[], int taille, int fx[]) {
    for (int i = 0; i < taille; i++) {
        fx[i] = x[i] * x[i] + 4 * x[i] + 1; // Calcul de f(x)
    }
}
```

8. b)

## → Solution :

```

#include <stdio.h>
#define MAX 10 // Taille maximale du tableau

// Fonction qui calcule  $f(x) = x^2 + 4x + 1$  pour un tableau donné
void calculF(int x[], int taille, int fx[]) {
    for (int i = 0; i < taille; i++) {
        fx[i] = x[i] * x[i] + 4 * x[i] + 1; // Calcul de f(x)
    }
}

int main() {
    int x[MAX], fx[MAX], taille;

    // Saisie de la taille du tableau
    printf("Entrez la taille du tableau (max %d) : ", MAX);
    scanf("%d", &taille);

    if (taille > MAX || taille <= 0) {
        printf("Erreur : La taille doit être entre 1 et %d.\n", MAX);
        return 1;
    }

    // Saisie des valeurs de x
    printf("Entrez %d valeurs pour x :\n", taille);
    for (int i = 0; i < taille; i++) {
        scanf("%d", &x[i]);
    }

    // Calcul des valeurs de f(x) en utilisant la fonction calculF
    calculF(x, taille, fx);

    // Affichage des résultats
    printf("Les valeurs de f(x) sont :\n");
    for (int i = 0; i < taille; i++) {
        printf("f(%d) = %d\n", x[i], fx[i]);
    }

    return 0;
}

```

## ➔ Sortie de Program :

```
Entrez la taille du tableau (max 10) : 4
Entrez 4 valeurs pour x :
3
4
2
3
Les valeurs de f(x) sont :
f(3) = 22
f(4) = 33
f(2) = 13
f(3) = 22
```

## 9. Écrire un programme qui permet la saisie et l'affichage des informations de plusieurs articles (struct Articles) : ?

### ➔ Solution :

```

1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. typedef struct {
5.     int code;
6.     char titre[100];
7.     char auteur[50];
8.     char editeur[50];
9.     int volume;
10.    char datePub[20];
11.} Article;
12.// Fonction pour saisir les informations d'un article
13.void saisirArticle(Article *a) {
14.    printf("\nEntrez le code de l'article : ");
15.    scanf("%d", &a->code);
16.    printf("Entrez le titre de l'article : ");
17.    scanf(" %[^\n]", a->titre); // %[^\n] pour lire toute la ligne
18.    printf("Entrez le nom de l'auteur : ");
19.    scanf(" %[^\n]", a->auteur);
20.    printf("Entrez le nom de l'Editeur : ");
21.    scanf(" %[^\n]", a->editeur);
22.    printf("Entrez le numero de volume : ");
23.    scanf("%d", &a->volume);
24.    printf("Entrez la date de publication (ex: JJ/MM/AAAA) : ");
25.    scanf(" %[^\n]", a->datePub);
26.}
27.

```



## ➔ Solution 9 (Suite):

```

28.
29.// Fonction pour afficher les informations d'un article
30.void afficherArticle(Article a) {
31.    printf("\n--- Informations de l'article ---\n");
32.    printf("Code      : %d\n", a.code);
33.    printf("Titre      : %s\n", a.titre);
34.    printf("Auteur     : %s\n", a.auteur);
35.    printf("Editeur    : %s\n", a.editeur);
36.    printf("Volume     : %d\n", a.volume);
37.    printf("Date Pub.   : %s\n", a.datePub);
38.}
39.
40.int main() {
41.    int n;
42.    printf("Entrez le nombre d'articles : ");
43.    scanf("%d", &n);
44.    Article *articles = (Article *)malloc(n * sizeof(Article));
45.    if (articles == NULL) {
46.        printf("Erreur d'allocation memoire.\n");
47.        return 1;
48.    }
49.    for (int i = 0; i < n; i++) {
50.        printf("\n--- Article %d ---\n", i + 1);
51.        saisirArticle(&articles[i]);
52.    }
53.    printf("\nListe des articles :\n");
54.    for (int i = 0; i < n; i++) {
55.        afficherArticle(articles[i]);
56.    }
57.    free(articles);
58.    return 0;
59.}
60.

```

## ➔ Sortie de Program :

```

Entrez le nombre d'articles : 3

--- Article 1 ---

Entrez le code de l'article : 2334
Entrez le titre de l'article : Programmation en C
Entrez le nom de l'auteur : Mohamed
Entrez le nom de l'Editeur : mohamed
Entrez le numero de volume : 4
Entrez la date de publication (ex: JJ/MM/AAAA) : 02/02/2002

--- Article 2 ---

Entrez le code de l'article : 190
Entrez le titre de l'article : Structure en C
Entrez le nom de l'auteur : Prof Asimi
Entrez le nom de l'Editeur : asimi
Entrez le numero de volume : 2
Entrez la date de publication (ex: JJ/MM/AAAA) : 11/04/2025

--- Article 3 ---

Entrez le code de l'article : 101
Entrez le titre de l'article : Bases de Donnes
Entrez le nom de l'auteur : Prof Lkermane
Entrez le nom de l'Editeur : lkeremane
Entrez le numero de volume : 5
Entrez la date de publication (ex: JJ/MM/AAAA) : 20/04/2025

Liste des articles :

--- Informations de l'article ---
Code      : 2334
Titre     : Programmation en C
Auteur    : Mohamed
Editeur   : mohamed
Volume    : 4
Date Pub. : 02/02/2002

--- Informations de l'article ---
Code      : 190
Titre     : Structure en C
Auteur    : Prof Asimi
Editeur   : asimi
Volume    : 2
Date Pub. : 11/04/2025

--- Informations de l'article ---
Code      : 101
Titre     : Bases de Donnes
Auteur    : Prof Lkermane
Editeur   : lkeremane
Volume    : 5
Date Pub. : 20/04/2025

```

## Exercice 10 :

### ➔ Solution :

```

1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. #define MAX 10 // Taille maximale
5.
6. // Déclaration de la structure
7. typedef struct {
8.     int nbElem;
9.     int *tabNotes;
10.    int *tabCoeff;
11.    int *tabMoy;
12.} StructTab;
13.
14.// Fonction pour calculer  $f(x) = x^2 + 4x + 1$ 
15.void calculF(int x[], int taille, int fx[]) {
16.    for (int i = 0; i < taille; i++) {
17.        fx[i] = x[i] * x[i] + 4 * x[i] + 1;
18.    }
19.}
20.
21.int main() {
22.    StructTab tab;
23.    int i;
24.
25.    // Saisie du nombre d'éléments
26.    printf("Entrez la taille du tableau (max %d) : ", MAX);
27.    scanf("%d", &tab.nbElem);
28.
29.    if (tab.nbElem > MAX || tab.nbElem <= 0) {
30.        printf("Erreur : La taille doit etre entre 1 et %d.\n", MAX);
31.        return 1;
32.    }
33.
34.    // Allocation dynamique
35.    tab.tabNotes = (int *)malloc(tab.nbElem * sizeof(int));
36.    tab.tabCoeff = (int *)malloc(tab.nbElem * sizeof(int));
37.    tab.tabMoy   = (int *)malloc(tab.nbElem * sizeof(int));
38.

```

## ➔ Solution 10 (Suite):

```

39.
40.     if (!tab.tabNotes || !tab.tabCoeff || !tab.tabMoy) {
41.         printf("Erreur d'allocation memoire.\n");
42.         return 1;
43.     }
44.
45.     // Saisie des notes
46.     printf("Entrez %d notes :\n", tab.nbElem);
47.     for (i = 0; i < tab.nbElem; i++) {
48.         printf("Note %d : ", i + 1);
49.         scanf("%d", &tab.tabNotes[i]);
50.
51.         // Coefficient croissant pour chaque note
52.         tab.tabCoeff[i] = i + 1;
53.     }
54.
55.     // Calcul des moyennes via f(x)
56.     calculF(tab.tabNotes, tab.nbElem, tab.tabMoy);
57.
58.     // Affichage
59.     printf("\n--- Resultats ---\n");
60.     for (i = 0; i < tab.nbElem; i++) {
61.         printf("Note : %d | Coeff : %d | Moyenne (f(x)) : %d\n",
62.             tab.tabNotes[i], tab.tabCoeff[i], tab.tabMoy[i]);
63.     }
64.
65.     // Libération mémoire
66.     free(tab.tabNotes);
67.     free(tab.tabCoeff);
68.     free(tab.tabMoy);
69.
70.     return 0;
71. }
72.

```

## ➔ Sortie de Program :

```
Entrez la taille du tableau (max 10) : 2
Entrez 2 notes :
Note 1 : 12
Note 2 : 9

--- Resultats ---
Note : 12 | Coeff : 1 | Moyenne (f(x)) : 193
Note : 9 | Coeff : 2 | Moyenne (f(x)) : 118
```

**Exercice 11 :**

Écrire un programme en langage C qui déclare un modèle d'une structure nommée *Student* , contenant les champs suivants : ?

**➔ Solution :**

```

1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. typedef struct {
5.     char nom[20];
6.     float* notes;
7.     float* moy;
8. } Student;
9.
10. int main() {
11.     int N, M;
12.     printf("Entrez le nombre d'etudiants : ");
13.     scanf("%d", &N);
14.
15.     // Allocation dynamique d'un tableau de N étudiants
16.     Student* etudiants = (Student*)malloc(N * sizeof(Student));
17.
18.     printf("Entrez le nombre de notes par etudiant : ");
19.     scanf("%d", &M);
20.
21.     // Allocation et initialisation
22.     for (int i = 0; i < N; i++) {
23.         printf("\nEtudiant %d\n", i + 1);
24.         printf("Nom : ");
25.         scanf("%s", etudiants[i].nom);
26.
27.         // Allocation des notes
28.         etudiants[i].notes = (float*)malloc(M * sizeof(float));
29.         // Allocation pour une seule moyenne par étudiant
30.         etudiants[i].moy = (float*)malloc(sizeof(float));
31.
32.         float somme = 0.0;
33.         for (int j = 0; j < M; j++) {
34.             printf("Note %d : ", j + 1);
35.             scanf("%f", &etudiants[i].notes[j]);
36.             somme += etudiants[i].notes[j];
37.         }
38.

```

**➔ Solution 11 (Suite):**

```
39.  
40.     *(etudiants[i].moy) = somme / M;  
41. }  
42.  
43. // Affichage des informations  
44. printf("\nAffichage des etudiants :\n");  
45. for (int i = 0; i < N; i++) {  
46.     printf("\nNom : %s\n", etudiants[i].nom);  
47.     printf("Notes : ");  
48.     for (int j = 0; j < M; j++) {  
49.         printf("%.2f ", etudiants[i].notes[j]);  
50.     }  
51.     printf("\nMoyenne : %.2f\n", *(etudiants[i].moy));  
52. }  
53.  
54. // Libération de la mémoire  
55. for (int i = 0; i < N; i++) {  
56.     free(etudiants[i].notes);  
57.     free(etudiants[i].moy);  
58. }  
59. free(etudiants);  
60.  
61. return 0;  
62.}  
63.
```

## ➔ Sortie de Program :

```
Entrez le nombre d'etudiants : 3
Entrez le nombre de notes par etudiant : 2

Etudiant 1
Nom : Ahmed
Note 1 : 12
Note 2 : 9

Etudiant 2
Nom : Youssef
Note 1 : 18
Note 2 : 20

Etudiant 3
Nom : Amal
Note 1 : 16
Note 2 : 16

Affichage des etudiants :

Nom : Ahmed
Notes : 12.00 9.00
Moyenne : 10.50

Nom : Youssef
Notes : 18.00 20.00
Moyenne : 19.00

Nom : Amal
Notes : 16.00 16.00
Moyenne : 16.00
```



**Exercice 12 :**

écrire un programme en C qui convertit un nombre entier positif en binaire (sur 2 et 4 octets) en suivant la structure du programme suivant ?

**➔ Solution :**

```

1. #include <stdio.h>
2. #include <stdbool.h>
3.
4. int main() {
5.     int rest, NombD, NombBin, j = 0, i = 0;
6.     bool TBin[32]; // tableau pour stocker les bits (max 32 bits)
7.
8.     printf("Saisir un nombre entier positif : ");
9.     scanf("%d", &NombD);
10.
11.    // Vérification du nombre (doit être positif)
12.    if (NombD < 0) {
13.        printf("Erreur : le nombre doit être positif.\n");
14.        return 1;
15.    }
16.
17.    // Conversion en binaire
18.    while (NombD > 0) {
19.        rest = NombD % 2;
20.        TBin[i] = rest;
21.        NombD = NombD / 2;
22.        i++;
23.    }
24.
25.    printf("Le nombre binaire est :\n");
26.    for (j = i - 1; j >= 0; j--) {
27.        printf("%d", TBin[j]);
28.    }
29.
30.    printf("\n");
31.
32.    return 0;
33.}
34.

```

**➔ Sortie de Program :**

```

Saisir un nombre entier positif : 3
Le nombre binaire est :
11

```