

**Université Moulay Ismail**  
**Faculté des Sciences et Techniques Errachidia**

**Filière : Cycle d'Ingénieur en Génie Informatique**

**Module**  
**Structure de données et programmation avancée en C**

**Compte rendu des Travaux Pratiques**

Réalisé par  
**AGDID MOHAMED**

Responsables  
**Prof. Y. FARHAOUI**  
**Prof. M. K. BOUTAHIR**

Année Universitaire : 2023/2024

## Table de Matière

|               |    |
|---------------|----|
| Série 1 ..... | 3  |
| Série 2 ..... | 10 |
| Série 3 ..... | 22 |
| Série 4 ..... | 29 |

## Série 1:

### Exercice 1:

```
#include <stdio.h>
#include<string.h>
int main()
{
    int x, y;
    char z;
    printf("saisie le nombre 1 : ");
    scanf("%d", &x);
    printf("\nsaisie le nombre 2 : ");
    scanf("%d", &y);

    printf("\nsaisie l'operateur : ");
    while(getchar() != '\n');
    scanf("%c",&z);
    switch (z){
        case '+': printf("\nresultat : %d", x + y); break;
        case '-': printf("\nresultat : %d", x - y); break;
        case '/': printf("\nresultat : %f", (float)x / y); break;
        case '*': printf("\nresultat: %d", (float)x * y); break;
        default: break;
    }
    return 0;
}
```

### Exercice 2:

```
#include <stdio.h>
int main() {
    int age;
    char sexe;
    printf("Veuillez entrer votre age: ");
    scanf("%d", &age);
    printf("Veuillez entrer votre sexe (H ou F) en majuscule : ");
    scanf(" %c", &sexe);
    if (sexe == 'H' && age > 20) {
        printf("L'habitant est imposable.\n");
    }
    else if (sexe == 'F' && age >= 18 && age <= 35) {
        printf("non, l'habitant est imposable.\n");
    }
    else {
        printf("oui, paient l'impôt.\n");
    }
}
```

```
return 0; }
```

### Exercice 3:

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

```
int main() {
```

```
int n;
```

```
float prix, total;
```

```
printf("Veuillez entrer le nombre de photocopies: ");
```

```
scanf("%d", &n);
```

```
if (n <= 10) {
```

```
    prix = 1.0;
```

```
} else if (n <= 30) {
```

```
    prix = 0.6;
```

```
} else {
```

```
    prix = 0.4;
```

```
}
```

```
total = n * prix;
```

```
printf("Le montant total pour %d photocopies est de %.2f  
DH.\n", n, total);
```

```
return 0;
```

```
}
```

### Exercice 4:

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

```
int main() {
```

```
float prix, prix_final, remise;
```

```
printf("Veuillez entrer le prix d'un produit : ");
```

```
scanf("%f", &prix);
```

```
if (prix < 100) {
```

```
    remise = prix * 0.3;
```

```
} else if (prix >= 100 && prix <= 200) {
```

```
    remise = prix * 0.4;
```

```
} else {
```

```
    remise = prix * 0.5;
```

```
}
```

```
prix_final = prix - remise;
```

```
printf("Prix initial : %.2f DH, Remise : %.2f DH, Prix final : %.2f  
DH.\n", prix, remise, prix_final);
```

```
return 0;
```

```
}
```

### Exercice 5:

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

```

int main() {
int n, i, ok = 0;
printf("Entrez un nombre entier : ");
scanf("%d", &n);
for (i = 2; i <= n/2 ;i++) {
3 if (n % i == 0) {
ok ++;
}
}
if (ok == 0) {
printf("%d est un nombre premier.\n",n);
}
else {
printf("%d n'est pas un nombre premier.\n", n);
}
return 0;
}

```

### Exercice 6:

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

```

int main(){
int n,i;
printf("Donner un nombre ");
scanf("%d",&n);
if(n%2==0){
printf("les nombres pairs qui lui sont inferieur sont: \n");

for(i=n;i>0;i--){
if(i%2==0) {
printf("%d \n",i);
}
}}
else{
printf("n'est pas pair.))}

```

### Exercice 7:

```

#include int main(){
//question 1
int nbr,i;

```

```

    for(i=1;i<=20;i++){
        printf("donner le nombre numero %d: ",i);
        scanf("%d",&nbr);
        if(nbr%2==0){
            printf("Le carre de %d est %d\n",nbr,nbr*nbr); }

    }
    // question 2
    int cmp=0, Som=0;
    do{
        printf("Entere un nombre : ");
        scanf("%d",&nbr);
        if(nbr!=100){
            if(nbr%2==0){
                printf("Le carre de %d est %d\n",nbr,nbr*nbr); }
            cmp++;
            Som+=nbr;}
        else{
            printf("fin");
        }while(nbr!=100);

    return 0;
}

```

### Exercice 8 :

```

#include <stdio.h>
int main() {
    int i,j,n;
    \A
    printf("enter un nombre: ");
    scanf("%d",&n);
    for(i=0;i<n;i++){
        for(j=n-i;j>0;j--){
            printf("%d",j);}
        printf("\n");}
    printf("\n");

    //B
    int k;
    for(i=0;i<n;i++){
        for(j=0;j<i;j++){
            printf(" ");

```

```

}
for(k=10;k>i;k--){
printf("%d",i);
}
printf("\n");
}
return 0;
}

```

### Exercice 9:

```

#include<stdio.h>
int main(){
int n, cmp=0, Som, Sompos, min, minpos;

do{
printf("Donner un nombre: ");
scanf("%d",&n);
if(n!=999){
cmp++;
min=n;
Som=n;
if(n>0){
minpos=n;
Sompos=n;}}
break;
}while(n!=999);

do{
printf("Donner un nombre: ");
scanf("%d",&n);
if(n!=999){
cmp++;
if(n<min){
min=n;}
Som+=n;
if(n>0){
if(n<min){
minpos=n;}
Sompos+=n;}
}}while(n!=999);

```

```

    printf("Le nombre total de valeurs de la
suite: %d\n",cmp);
    printf("La somme des valeurs lues: %d\n",Som);
    printf("Le minimum: %d\n",min);
    printf("La somme des valeurs strictement
positives: %d\n",Sompos);
    printf("Le minimum des valeurs strictement
positives: %d\n",minpos);

    return 0;

}

```

### Exercice 10:

```

#include<stdio.h>
int main(){
    int i,j,n;
    printf("Saisir nombre des lignes: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        for(j=1;j<2*n;j++){
            if(j==n-(i-1)|| j==n+(i-1)|| i==n)    printf("*");
            else printf(" ");
        }
        printf("\n");
    }
}

```

### Exercice 11:

```

#include <stdio.h>
void option(int n){
    int choix;
    printf("Faire autre option : 1-OUI / 2-NON");
    scanf("%d",&choix);
    if(choix == 1){
        MENU(n);}
    else(
        exit(0);}
}
void MENU(int n) {
    int x;

```



```

printf("_____ \n");
printf("|__MENU_____ \n");
printf("| [1] Ajouter 2 _____ |\n");
printf("| [2] Multiplier par 3 _____ |\n");
printf("| [3] Soustraire 5 _____ |\n");
printf("| [4] Quitter _____ |\n");
printf("|_____ |\n");
    printf("Votre choix: ");
    scanf("%d", &x);
    switch(x){
    case 1: printf("%d\n",n+2); break;
    case 2: printf("%d\n",n*3); break;
    case 3: printf("%d\n",n-5); break;
    case 4: exit(0); break;}
    option(n);
}

int main(){
int n;
printf("taper un nombre:");
scanf("%d",&n);
return 0;
}

```

### Exercice 12:

```

#include<stdio.h>
int main(){
    int i,j,n ;
    printf("Saisir la taille: ");
    scanf("%d",&n);
    for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        if(i==0 || i==n-1 || j==n-1 || j==0){
            printf("*");}
        else if(j == n/2){
            printf("-");}
        else if(i == n/2 && j!=n/2){

```

```

        printf("|");}
    else if ( (i==j && i!=0 & i!=n-1 ) || (i == j-n-1 && j!=n/2 ) ){
        printf("+");}
    else if ((i == n - 1 - j && i != 0 && i != n - 1) || (i == j && j != n
/ 2)) {
        printf("*"); }
    else{
        printf(" ");}}
    printf("\n");
}
return 0;}

```

## serie2:

### Exercice1:

```

#include <stdio.h>
#define Nmax 20

void Saisie(int T[]) {
    int N, i;
    do {
        printf("Enter nombre des elements (<= %d): ", Nmax);
        scanf("%d", &N);
    } while (N > Nmax);

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

void Affichage(int T[]) {
    int i;
    for (i = 0; i < Nmax; i++) {
        printf("%d\t", T[i]);
    }
    printf("\n");
}

```

```

void Moyenne(int T[]) {
    int som = 0, i;
    float moy;
    for (i = 0; i < Nmax; i++) {
        som += T[i];
    }
    moy = (float)som / Nmax;
    printf("La moyenne est : %.2f\n", moy);
}

```

```

int Max_elem(int T[]) {
    int max = T[0], pos_max = 0, i;
    for (i = 1; i < Nmax; i++) {
        if (T[i] > max) {
            max = T[i];
            pos_max = i;
        }
    }
    return pos_max;
}

```

```

void Supprime_max(int T[]) {
    int pos = Max_elem(T);
    int i;
    for (i = pos; i < Nmax - 1; i++) {
        T[i] = T[i + 1];
    }
    Nmax--;
}

```

```

int Min_elem(int T[]) {
    int min = T[0], pos_min = 0, i;
    for (i = 1; i < Nmax; i++) {
        if (T[i] < min) {
            min = T[i];
            pos_min = i;
        }
    }
    return pos_min;
}

```

```

void Supprime_min(int T[]) {

```

```

    int pos = Min_elem(T);
    int i;
    for (i = pos; i < Nmax - 1; i++) {
        T[i] = T[i + 1];
    }
    Nmax--;
}

void Ajout(int T[]) {
    int posADD, i;
    int n;
    do {
        printf("Enterez la position (<= %d): ", Nmax);
        scanf("%d", &posADD);
    } while (posADD > Nmax || posADD < 0);

    printf("Enter nombre a ajouter: ");
    scanf("%d", &n);
    Nmax++;

    for (i = Nmax - 1; i > posADD; i--) {
        T[i] = T[i - 1];
    }
    T[posADD] = n;
}

```

```

int main() {
    int T[Nmax];
    int x;

    printf("_____ \n");
    printf("|__ MENU _____ \n");
    printf("|[1] Saisie des elements | \n");
    printf("| [2] Moyenne | \n");
    printf("| [3] Supprime max | \n");
    printf("| [4] Supprime min | \n");
    printf("| [5] Ajouter elements | \n");
    printf("|_____ | \n");
    printf("Votre choix: ");
    scanf("%d", &x);
    switch (x) {
    case 1:
        Saisie(T);

```

```

        Affichage(T);
        break;
    case 2:
        Moyenne(T);
        break;
    case 3:
        Supprime_max(T);
        Affichage(T);
        break;
    case 4:
        Supprime_min(T);
        Affichage(T);
        break;
    case 5:
        Ajout(T);
        Affichage(T);
        break;
    default:
        break;
}
return 0;
}

```

## Exercice2:

```

#include<stdio.h>

int i,j;
void Saisie(int iMat[5][5]){
    for(i=0; i<5; i++)
        for(j=0; j<5; j++){
            printf("iMat[%d][%d] = ",i,j);
            scanf("%d",&iMat[i][j]);
        }
}

void Afficher(int iMat[5][5]){
    for(i=0; i<5; i++){
        for(j=0; j<5; j++)
            printf("| %d ",iMat[i][j]);
        printf("\n");
    }
}

main(){

```

```

int iMat[5][5];
Saisie(iMat);
Afficher(iMat);
}

```

### Exercice3:

```

#include<stdio.h>

```

```

void Initialisation(int iNb_jours[]){
    inti;
    for(i=0;i<12;i++){
        if(i==2) iNb_jours[i] = 28;
        else if( (i%2==0 && i<=7) || (i%2!=0 && i>7)) iNb_jours[i] = 30;
        else iNb_jours[i] = 31;
    }
}

```

```

void mois(int iNb_jours[]){
    int n;
    printf("Donner un mois: ");
    scanf("%d",&n);
    printf("Nmb de jours est = %d\n",iNb_jours[n]);
}

```

```

void Affichage(int iNb_jours[]){
    int i;
    for(i=1;i<=12;i++)
        printf("iNb_jours[%d] = %d\n",i,iNb_jours[i]);
}

```

```

int main(){
    int iNb_jours[12];
    initialisation(iNb_jours);
    Affichage(iNb_jours);
    mois(iNb_jours);
    return 0;
}

```

### Exercice 4;

```

#include<stdio.h>
#include<string.h>
int longueur_chaine1(char C[]){
int taille=0,i=0;

while(T[i]!='\0'){
taille++;
i++;
}
return taille;
}

int main(){
char Ctab1[20],Ctab2[20];
printf("tapez chaine 1");
gets(Ctab1);
printf("tapez chaine 2");
gets(Ctab2);
printf("taille de chaine 1 est : %d", longueur_chaine1(Ctab1));
printf("\ntaille de chaine 2 est : %d", longueur_chaine1(Ctab2));
return 0;
}

```

### Exercice5:

```

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

void cryptage(char KRP[]) {
    int i;

    for (i = 0; i < strlen(KRP); i++) {
        if ((KRP[i] >= 'A' && KRP[i] <= 'Z') || (KRP[i] >= 'a' &&
KRP[i] <= 'z')) {
            KRP[i] += 5;
        }
    }
    puts(KRP);
}

```

```

int main() {
    char CH[20];

    printf("Donner la chaine a crypter: ");
    scanf(" %s", CH);

    printf("avant le cryptage est: %s\n", CH);
    printf("apres le cryptage est: ");
    cryptage(CH);

    return 0;
}

```

### Exercice 6:

```

#include<stdio.h>
#include<string.h>
#include <ctype.h>
int main(){
    int i;
    int nbrEsp=0, nbrChif=0, nbrAut=0;
    char T[50];

    printf("Saisir un chaine de caractere: ");
    gets(T);

    for(i=0;i<strlen(T);i++){
        if(T[i] == ' ' || T[i] == '\n' || T[i] == '\t') nbrEsp++;
        else if(isdigit(T[i])) nbrChif++;
        else nbrAut++;
    }
    printf("Nombres des espaces, des tabulations et des retours
a la ligne est: %d\n",nbrEsp);
    printf("Nombres des chiffres est: %d\n",nbrChif);
    printf("Nombres des autres caracteres est: %d",nbrAut);
    return 0;
}

```



## Exercise 7:

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

void Addition(int A, int B){
printf("%d + %d = %d\n", A,B,A+B);
}
void Soustraction(int A, int B){
printf("%d - %d = %d\n", A,B,A-B);
}
void Multiplication(int A, int B){
printf("%d * %d = %d\n", A,B,A*B);
}
void Division(int A, int B){
printf("%d / %d = %d\n", A,B,A/B);
}
void Modulo(int A, int B){
printf("%d % %d = %d\n", A,B,A%B);
}

int main (){
    int A,B;
    char x,y;
    printf("enter A et B : ");
    scanf ("%d %d", &A,&B);
    do{
        printf("Enter votre operation (+,-,*,/,%) :");
        scanf("%c",&x);
        switch (x){
            case '+': Addition(A,B); break;
            case '-': Soustraction(A,B); break;
            case '*': Multiplication(A,B); break;
            case '/': Division(A,B); break;
            case '%': Modulo(A,B); break;
            default : break;}
        printf("\ncontinuer:\n O: Oui \t N: Non  ");
        scanf(" %c",&y);
    }while( y == 'O');
```

```
return 0;
}
```

### Exercice 8:

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

int comparaison(char* s,char* t){
    return strcmp(s,t);
}

int Saisire(char* nom){
    printf("Donnez un nom (Entrez fin si vous avez terminer) : ");
    scanf("%s", nom);
    return strlen(nom);
}

int main(){
    char nom[20];
    int nomlen,cmp=0;

    do{
        int nomlen = Saisire(nom);
        if(nomlen>10){
            cmp++;
        }
    }while(comparaison(nom,"fin") != 0);
    printf("\nLe nombre de mots dont une taille plus de 10
est : %d",cmp);
    return 0;
}
```

### Exercice 9:

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

```

int MENU() {
    int x;

    printf("_____ \n");

    printf("|__ MENU_____ \n");
    printf("| [1] Saisir | \n");
    printf("| [2] Affichage | \n");
    printf("| [3] Inverser | \n");
    printf("| [4] Compter les mots | \n");
    printf("| [5] Quitter | \n");
    printf("_____ | \n");
    printf("Votre choix: ");
    scanf("%d", &x);
    return x;
}

char* Saisie() {
    char *ch = (char*)malloc(100 * sizeof(char));
    printf("\nDonnez la chaine : ");
    scanf("%s", ch);
    return ch;
}

void Affichage(char* s) {
    printf("%s", s);
}

void Inverser(char* T) {
    int i, j;
    char ch[strlen(T)];
    strcpy(ch, T);

    for (i = 0; i < strlen(T); i++) {
        T[i] = ch[strlen(T) - 1 - i];
    }
    printf("L'inverse de votre chaine est : %s\n", T);
}

void Cmptmots(char* T) {
    int i, cmp = 0;
    for (i = 0; i < strlen(T); i++) {
        if (T[i] == ' ' || T[i] == '\t' || T[i] == '\n') {

```

```

        cmp++;
    }
}
printf("Le nombre de mots sans espaces dans votre
chaîne est : %d\n", strlen(T) - cmp);
}

```

```

int main() {
    char CH[100];
    int x = MENU();

    switch (x) {
        case 1:
            strcpy(CH, Saisie());
            break;
        case 2:
            Affichage(CH);
            break;
        case 3:
            Inverser(CH);
            Affichage(CH);
            break;
        case 4:
            Cmptmots(CH);
            Affichage(CH);
            break;
        case 5:
            exit(0);
            break;
        default:
            break;
    }
    return 0;
}

```

### exercice 10:

```

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

```

```

int distanceH(char* s1, char* s2){
    int i, nbr=0;

```

```

for(i=0;i<strlen(s1);i++){
    if(s1[i] != s2[i]){
        nbr++;
    }
}
return nbr;
}

```

```

int distanceH_langage(char** langage,int t){
    int i,min=100,H;
    for(i=0;i<t-1;i++){
        H = distanceH(langage[i],langage[i+1]);
        if(min>H){
            min = H;
        }
    }
    return min;
}

```

```

int* binaire(int nbrdec){
    int T[8],i;
    for(i=7;i>=0;i--){
        T[i] = nbrdec%2;
        nbrdec = nbrdec/2;
    }
    return T;
}

```

```

int distanceH_entier(int N1,int N2){
    int i,*T1,*T2,cmp=0;
    T1=binaire(N1);
    T2=binaire(N2);
    for(i=0;i<8;i++){
        if(T1[i] != T2[i]){
            cmp++;
        }
    }
    return cmp;
}

```

```

int main(){
    char S1[10],S2[10];
    int i;

```

```

// 1
do{
    printf("Donnez S1 : ");
    gets(S1);
    printf("Donnez S2 : ");
    gets(S2);
}while(strlen(S1) != strlen(S2));
printf("La distance de Hamming entre '%s' et '%s'
est : %d",S1,S2,distanceH(S1,S2));
// 2
    int taille;

printf("\nDonnez le nombre de mots : ");
    scanf("%d",&taille);
char **lang = (char **)malloc(taille*sizeof(char*));
for (i=0;i<taille;i++){
    lang[i] = (char *) malloc (50*sizeof(char));
}
    printf("Donnez le langage : \n");
    for(i=0;i<taille;i++){
        printf("mot %d : ",i+1);
        scanf("%s",lang[i]);
    }
    printf("La distance de Hamming de langage
est : %d",distanceH_langage(lang,taille));

// 3
int n1,n2;
printf("\nDonnez nombre : ");
    scanf("%d",&n1);
    printf("\nDonnez autre nombre : ");
    scanf("%d",&n2);
    printf("La distance de Hamming entre les nombres %d et %d
est : %d",n1,n2,distanceH_numbers(n1,n2));

return 0;
}

```

## Serie 3:

### exercice 1:

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

int main() {
    char T[30];
    char *q, *p, temp;
    printf("Tapez votre mot : ");
    gets(T);
    q = T;
    p = &T[strlen(T) - 1];

    for(q,p;p>q;p++,q++){
        temp = *q;
        *q = *p;
        *p = temp;
    }

    for(q;*q;q++){
        printf("%c",*q);
    }
    return 0;
}
```

### Exercice 2:

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

int main(){
    int i=0,taille;
    char Tab1[50],Tab2[50];

    printf("Tapez un mot:");
    scanf("%s",Tab1);

    while(Tab1[i] != '\0'){
        i++;
    }
    taille= i-1;
    for(i=0;i< taille ;i++){
```

```

        Tab2[i]=Tab1[taille-1-i];
    }
    int OK=0;
    for(i=0;i<taille ; i++){
        if(Tab1[i] == Tab2[i]){
            OK==1;}
        else{OK==0;}}

    if (OK==1){printf("Le mot saisis est palindrome");}
    else {printf("Le mot saisis n'est pas palindrome);}
    return 0;
}

```

### Exercie 3:

```

#include<stdio.h>
#include<stdlib.h>
#define n 3
#define m 4

int i,j;
float Somme_usuel(float t[3][4]){
    float Som=0;
    for(i=0; i<3; i++) {
        for(j=0; j<4; j++) {
            Som += t[i][j];
        }
    }
    return Som;
}

float Somme_pointeur(float **t,int n, int m){
    int i,j;
    float Somm=0;
    for(i=0;i<n;i++){
        for (j=0 ; j<m;j++){
            Somm+= *(t+i*m+j);}}
    return Somm;
}

int main(){
    int i,j;

```



```

float t[3][4];
float Somme;

for(i=0; i<3; i++){
for(j=0; j<4; j++){
    printf("t[%d][%d] = ",i,j);
    scanf("%f",&t[i][j]);}
float **T;
T=(float**) malloc(n*sizeof(float*));
for(i=0;i<n;i++){
    T[i]=(float*)malloc(m*sizeof(float));}

//Remplissage
for(i=0; i<3; i++){
for(j=0; j<4; j++){
*(T+i*m+j)=t[i][j];
}}

// Affchage
for(i=0; i<3; i++){
for(j=0; j<4; j++) {
printf("%.2f \t",*(t+i*m+j));}
printf("\n");
}

//Somme usuel
Somme=Somme_usuel(t);
printf("\nLa somme des elements du tableau
est: %.2f \n",Somme);

//Somme en pointeur
Somme=Somme_pointeur(T,n,m);
printf("\nLa somme des elements du tableau est: %.2f
\n",Somme);

return 0;
}

```

## Exercice 4:

1-

|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  |
| 5  | 6  | 7  | 8  | 9  |
| 10 | 11 | 12 | 13 | 14 |

2-  $^{**}(b+1) = 16; :$

|    |    |    |    |    |
|----|----|----|----|----|
| 15 | 16 | 3  | 4  | 5  |
| 6  | 7  | 8  | 9  | 10 |
| 11 | 12 | 13 | 14 | 15 |

3-  $^{*}(b[0]+1) = 17; :$

|    |    |    |    |    |
|----|----|----|----|----|
| 15 | 17 | 4  | 5  | 6  |
| 7  | 8  | 9  | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 |

4-  $^{*}(*b+8) = 18; :$

|    |    |    |    |    |
|----|----|----|----|----|
| 15 | 17 | 5  | 6  | 7  |
| 8  | 9  | 10 | 18 | 11 |
| 12 | 13 | 14 | 15 | 16 |

5-  $^{*}(b[1]+2) = 19; :$

|    |    |    |    |    |
|----|----|----|----|----|
| 15 | 17 | 19 | 6  | 7  |
| 8  | 9  | 10 | 18 | 11 |
| 12 | 13 | 14 | 15 | 16 |

6-  $^{*}(* (b+1)+5) = 20; :$

|    |    |    |    |    |
|----|----|----|----|----|
| 15 | 17 | 19 | 6  | 7  |
| 8  | 9  | 10 | 18 | 11 |
| 12 | 13 | 14 | 15 | 20 |

7-  $^{*}(b[2]+3) = 21; :$

|    |    |    |    |    |
|----|----|----|----|----|
| 15 | 17 | 19 | 6  | 7  |
| 8  | 9  | 10 | 18 | 11 |
| 12 | 13 | 14 | 21 | 20 |

8-  $^{*}(* (b+2)+2) = 22; :$

|    |    |    |   |   |
|----|----|----|---|---|
| 15 | 17 | 19 | 6 | 7 |
|----|----|----|---|---|

|    |    |    |    |    |
|----|----|----|----|----|
| 8  | 9  | 10 | 18 | 11 |
| 12 | 22 | 14 | 22 | 20 |

### Exercise 5:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int nbr_occ(char *ph, char *mot){
int resultat;
char *p=ph;
while( (p=strstr(p,mot)) != NULL){
    resultat++;
    p += strlen(mot);
}
    return resultat;
}

int main(){
    char *ph, *mot;
    int resultat;
    ph = (char*)malloc(250*sizeof(char));
    mot = (char*)malloc(15*sizeof(char));
    printf("Donner une phrase: ");
    printf("Donner un mot: ");
    gets(mot);
    resultat = nbr_occ(ph,mot);
    printf("\nLe nombre d'occurence est: %d",mot,resultat);
}
```

### Exercise 6:

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

void main(){
    char phrase[100];
    char phraseB[100];
    int nbrocc,ok=0;
    char mot[20];
    printf("Donner une phrase: ");
    gets(phrase);
    int i=0;
    while(phrase[i] != "){
        phraseB[i]=phrase[i];
```

```

        i++;
    }
    printf("\n%s",phraseB);
}

```

### Exercice 7:

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

```
int search(float T[], int n, float nbr){
```

```
    int i;
    int cmp=0;
    for(i=0; i<n; i++)
    if(T[i] == nbr){
        cmp++;
        break;}

```

```
    if(cmp==0) {return(-1)};
    }
    else{return i;}

```

```
int main(){
```

```
    int n;
    float x;
    printf("Donner taille de tableau: ");
    scanf("%d",&n);
    float T[n];

```

```
    for(i=0; i<n; i++){
        printf("T[%d] = ",i);
        scanf("%f",&T[i]);
    }

```

```
    printf("\nDonner la valeur a rechercher: ");
    scanf("%f",&x);
    int pos;
    pos=search(T,n,x);
    if(pos!=-1){
        printf("\nla position dans le tableaux est %d",pos);}
    else{printf("\n n'existe pas dans le tableaux);}

```

```
}
```

### Exercice 8:

```
#include<stdio.h>
#include<string.h>
void main(){
    int T[100], V[100];
    int taille, k=0;
    printf("Enter taille de tableaux : ");
    scanf("%d",&taille);
    printf("Enter l'element de tableaux : ");
    for(i=0;i<taille;i++){
        printf("T[%d]= ",i);
        scanf("%d",&T[i]); }
    for(i=0;i<taille;i++){
        ok=0;
        for(j=i;j<taille ; j++){
            if(T[i]==T[j]){
                ok++;
            }
            else{
                V[k]=ok;
                V[k+1]=T[i];
                k++;
                i+=ok;
            }
        }
    }
}
```

## serie 4:

### Exercice 1:

```
#include<stdio.h>

typedef struct complex{
    int re;
    int img;
}complex;
```

// passage par valeur

```
double imag(Complexe z){
    return z.img;
}
double reel(Complexe z){
    return z.re;
}
```

```
complex mul(complex z1, complex z2){
    complex Z;
    Z.re = z1.re*z2.re + z1.img*z2.img;
    Z.img = z1.re*z2.img + z1.img*z2.re;
    return Z;}

```

// passage par adresse

```
double* imag(Complexe *z){
    return *z.img;
}
double* reel(Complexe *z){
    return *z.re;
}
```

```
complex* mul(complex *z1, complex *z2){
    complex *Z=(complex*)malloc(sizeof(complex));
    Z->re = z1->re*z2->re + z1->img*z2->img;
    Z->img = z1->re*z2->img + z1->img*z2->re;

    return Z;}

```

```
void main(){
    Complexe z1, z2, z;
    printf("Donner la partie real de z1: ");
    scanf("%d",&z1.re);
    printf("Donner la partie imaginaire de z1: ");
    scanf("%d",&z1.img);
    printf("Donner la partie real de z2: ");
    scanf("%d",&z2.re);
    printf("Donner la partie imaginaire de z2: ");
}
```

```

scanf("%d",&z2.img);
printf("z1 = %.2lf + %.2lf i\n", reel(z1), imag(z1) );
printf("z2 = %.2lf + %.2lf i\n", reel(z2), imag(z2) );
z = mul(z1,z2);
printf("z1 x z2 = %.2lf + %.2lf i", reel(z), imag(z));
}

```

## Exercice 2:

```

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

typedef struct date{
    int jour;
    char mois[10];
    int annee;
}date;

typedef struct employe{
    char nom[15], prenom[15];
    date date_naissance, date_embauche;
}employe;

void main(){
    employe E[4];

    for(i=0;i<4;i++){
        printf("enter les infos d'employer %d :",i+1);
        printf("Nom:");
        scanf("%s",&E[i].nom);
        printf("Prenom:");
        scanf("%s",&E[i].prenom);
        printf("Date de Naissance:");
        printf("Jour:");
        scanf("%d",&E[i].date_naissance.jour);
        printf("Mois:");
        scanf("%s",&E[i].date_naissance.mois);
        printf("Annee:");
        scanf("%d",&E[i].date_naissance.annee);
        printf("Date d'embauche:");
        printf("Jour:");
    }
}

```

```

scanf("%d",&E[i].date_embauche.jour);
printf("Mois:");
scanf("%s",&E[i].date_embauche.mois);
printf("Annee:");
scanf("%d",&E[i].date_embauche.annee);
}

for(i=0;i<4;i++){
printf("Nom: %s",E[i].nom);
printf("Prenom: %s",E[i].prenom);
printf("Date                                     de
Naissance: %d %s %d",E[i].date_naissance.jour,E[i].date_nai
ssance.mois,E[i].date_naissance.annee);
printf("Date
d'embauche: %d %s %d",E[i].date_embauche.jour,E[i].date_e
mbauche.mois,E[i].date_embauche.annee);
}

```

### Exercice 3:

```

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

```

```

typedef struct etudiant {
    char nom[15], prenom[15];
    int CNE;
    float notes[4], moyenne;
} etudiant;

```

```

int main() {
    int n, i, j;
    float som = 0;
    etudiant temp;
    printf("Entrez le nombre d'étudiants : ");
    scanf("%d", &n);
    etudiant E[n];
    for (i = 0; i < n; i++) {
        printf("\nEntrez les informations pour l'étudiant %d :",
i + 1);
        printf("\nNom : ");
        scanf("%s", E[i].nom);
        printf("Prénom : ");
        scanf("%s", E[i].prenom);
    }
}

```



```

printf("CNE : ");
scanf("%d", &E[i].CNE);
printf("Entrez les notes : ");
for (j = 0; j < 4; j++) {
    printf("\nNote %d : ", j + 1);
    scanf("%f", &E[i].notes[j]);
    som += E[i].notes[j];}
E[i].moyenne = som / 4;
som = 0; }
for (i = 0; i < n - 1; i++) {
    for (j = 0; j < n - i - 1; j++) {
        if (E[j].moyenne < E[j + 1].moyenne) {
            temp = E[j];
            E[j] = E[j + 1];
            E[j + 1] = temp;}}}
printf("\nListe triée par moyenne :\n");
for (i = 0; i < n; i++) {
    printf("\nNom : %s", E[i].nom);
    printf("\nPrénom : %s", E[i].prenom);
    printf("\nCNE : %d", E[i].CNE);
    printf("\nNotes : ");
    for (j = 0; j < 4; j++) {
        printf("\nNote %d : %.2f", j + 1, E[i].notes[j]);}
    printf("\nMoyenne : %.2f\n", E[i].moyenne);}
return 0;}

```

#### Exercise 4:

```

#include<stdio.h>
typedef struct Repertoire{
    char nom[15];
    char prenom[15];
    int Tele;
}Repertoire;

void Afficher(Repertoire T[], int t){
    int i, j;
    printf("Les donnees saisis sont: \n");
    for(i=0; i<t; i++){
        printf("Repertoire %d \n",i+1);
        printf("Nom: %s \n",T[i].nom);
        printf("Prenom: %s \n",T[i].prenom);
        printf("Tele: %d \n",T[i].Tele);
    }
}

```

```
}
```

```
int main(){
int i, j, n;
printf("Donner le nombre des enregistrements : ");
scanf("%d",&n);
Repertoire T[n];

printf("\nEnter vos donnees");
for(i=0; i<n; i++){
printf("Enregistrement %d \n",i+1);
printf("Nom: ");
scanf(" %s",T[i].nom);
printf("Prenom: ");
scanf(" %s",T[i].prenom);
printf("Tele: ");
scanf("%d",&T[i].Tele);
printf("\n");
}
Afficher(T,n);
return 0;
}
```

### Exercise 5:

```
#include<stdio.h>
#include<string.h>
int nb_occurrence(char T[], int n, char c){
    int i, cmp=0;
    int p_pos, d_pos;
    for(i=0;i<n;i++){
        if (T[i]== c){
            p_pos=i;
            break;}}
    printf("p_occ = %d \n", p_pos);
    for(i=0;i<n;i++){
        if (T[i]== c){
            d_pos=i;
            cmp++;
        }}
    printf("d_occ = %d \n", d_pos)
```

```

        return cmp;}

int main(){

    int n;
    printf("Donner la taille du chaine : ");
    scanf("%d",&n);
    char Ch[n],c;
    printf("taper une phrase : ");
    gets(Ch);
    printf("Donner le caracter a cherche: ");
    scanf("%c",&c);
    printf("le nombre des occurences est : %d",
nb_occurrence(Ch,n,c));
    return 0;
}

```

### exercice 6:

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

```

typedef struct panneau{
    float l;
    float L;
    flaot ml;
    int type_bois;
}Panneau;

```

```

void Saisie(Panneau P[], int n){
    int i;
    printf("\nSaisir les informations des panneaux: \n");
    for(i=0; i<n; i++){
        printf("Panneau %d\n",i+1);
        printf("\nLargeur: ");
        scanf("%f",&P[i].l);
        printf("\nLongeur: ");
        scanf("%f",&P[i].L);
        printf("Epaisseur: ");
        scanf("%f",&P[i].ml);
        printf("Type du bois: \n 0: Pin \n 1: Chene \n 2: Hetre \n");
        scanf("%d",&P[i].type_bois); }
}

```

```

void Afficher(Panneau P[], int n){
    int i;
    printf("\n Les informations saisis sont : ");
    for(i=0; i<n; i++){
        printf("\n Panneau %d \n",i+1);
        printf("Largeur: %.2f mm \n",P[i].l);
        printf("Longueur: %.2f mm \n",P[i].L);
        printf("Epaisseur: %.2f mm \n",P[i].ml);
        printf("Type du bois: ");
        if(P[i].type_bois == 0) printf("Pin \n");
        else if(P[i].type_bois == 1) printf("Chene \n");
        else if(P[i].type == 2) printf("Hetre \n");
    }
}

float Volume(Panneau P){
    float V;
    V = P.l * P.L * P.ml;
    return V;
}

int main(){
    int n,i;
    printf("Donner le nombre des panneaux: ");
    scanf("%d",&n);
    Panneau P[n];
    Saisie(P,n);
    Afficher(P,n);
    printf("Donner le numero de panneau souhaiter
pour calculer le volume: ");
    scanf("%d",&i);
    printf("Le Volume est: %.2f m^3 \n", Volume(T[i-
1])) );
}

```

### Exercice 7:

```
include<stdio.h>
```

```

typedef struct Produit{
    int Ref;
    int Type;
    int Qte;

```

```

float Prix;
}Produit;

void Saisie(Produit P[]){
    printf("Veuillez saisir les donnees: \n");
    for(i=0; i<4; i++){
        printf("\n Produit %d \n",i+1);
        printf("Reference: ");
        scanf("%d",&P[i].Ref);
        printf("Type du produit: \n 1: Cartes meres \n 2: Processeurs
\n 3: Barettes memoire \n 4: Carte graphique \n");
        scanf("%d",&P[i].Type);
        printf("Quantite: ");
        scanf("%d",&P[i].Qte);
        printf("Prix: ");
        scanf("%f",&P[i].Prix);}

void Affichage(Produit P[]){
    for(i=0; i<4; i++){
        printf("\n Produit %d \n",i+1);
        printf("Reference: %d \n",P[i].Ref);
        printf("Type du produit: ");
        if(P[i].Type == 1) printf("Carte mere \n");
        else if(P[i].Type == 2) printf("Processeur \n");
        else if(P[i].Type == 3) printf("Barrettes memoire \n");
        else if(P[i].Type == 4) printf("Cartes graphique \n");
        printf("Quantite: %d \n",P[i].Qte);
        printf("Prix: %.2f DH \n",P[i].Prix);
    }
}

void Commande(Produit P[]){
    int type, qte;
    printf("\nSaisir votre commande SVP: \n");
    printf("Quel est le produit que vous voulez ? \n 1:
Cartes memoires \n 2: Processeurs \n 3: Barettes memoire \n
4: Carte graphique \n");
    scanf("%d",&type);
    printf("Quelle est la quantite ? \n");
    scanf("%d",&qte);
    for(i=0;i<4;i++){
        if( i == type){
            if(T[i].Qte < qte){
                printf("Qte n'eat pas disponible !!"); }
            printf("Le produit demander est: ");

```

```

        if(T[i].Type == 1) printf("Carte mere\n");
        else if(T[i].Type == 2) printf("Processeur \n");
        else if(T[i].Type == 3) printf("Barrettes memoire \n");
        else if(T[i].Type == 4) printf("Cartes graphique \n");
        printf("La Reference du produit: %d \n", T[i].Ref);
        printf("Le prix totoal est: %.2f DH \n", qte*T[i].Prix);}}
    }

int main(){
    Produit T[4];
    Saisie_Affichage(T);
    printf("Voulez-vous effectuer une commande ?
1:OUI 0:NON \n");
    scanf("%d",&i);
    if(i==1){ Commande(T);}
    return 0;}

```

### Exercice 8 :

```

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

typedef struct{
    int nb_elem;
    int *tab;
}TypeTableau;

TypeTableau CreationTableau(int n){
    TypeTableau T;
    T.nb_elem = n;
    T.tab = (int*)malloc(T.nb_elem * sizeof(T));
    return T; }

void DestructionTableau(TypeTableau T){
    free(T.tab);
}

void SimpleLectureTableau(TypeTableau T){
    int i;
    printf("Entere les element de tableaux");
    for(i=0;i<T.nb_elem;i++){
        printf("T[%d] = ",i);
        scanf("%d",&T.tab[i]);}}

void Affichage(TypeTableau T){

```

```

    int i;
    for(i=0;i<T.nb_elem;i++){
        printf("T[%d] = %d\n",i,T.tab[i]);
    }
}

```

```

TypeTableau DoubleTableau(TypeTableau T){
    for(i=0;i<T.nb_elem;i++){
        T.tab[i]=T.tab[i]*T.tab[i];}
}

```

```

int main(){
    int n;
    printf("Enter nombre des elements : ");
    scanf("%d",&n);
    TypeTableau M;
    M=CreationTableau(int n);
    SimpleLectureTableau(M);
    Affichage(A);
    M=DoubleTableau(M);
    printf("/nAprès doublication de matrice");
    Affichage(M);
    DestructionTableau(M);
    return 0;}

```

**FIN**