

Exercices 1

- 1) Soit le programme suivant :

```

int fonction1 (int i) { return ( i % 2 ); }
int fonction2 (int i) { return ( + + i ); }
int main () {
    int a , b;
    a = fonction1 ( 5 );
    b = fonction2 ( 1 );
    printf ("a = %d, b = %d\n", a, b)    }

```

L'exécution de ce programme affichera :

- A. a = 2, b = 1
 B. a = 2, b = 2
 C. a = 1, b = 1
 D. a = 1, b = 2
- 2) Ecrire un programme qui lit un fichier et l'imprime à l'écran, en remplaçant tous les chiffres par le symbole *. On rappelle que le caractère de fin de fichier est la constante *EOF* définie dans la librairie standard *stdio.h*
- 3) Ecrire un programme qui calcule et affiche la distance *DIST* (type *double*) entre deux points *A* et *B* du plan dont les coordonnées (*XA, YA*) et (*XB, YB*) sont entrées au clavier comme *entiers*.

Exercices 2 :Etant donné un tableau T du type **int** de dimension *n* (**int**). Ecrire les fonctions suivantes :

```

/**Remplissage du tableau**/
void saisie(int t[],int n){ .....}
/**Affichage des éléments du tableau**/
void affichage(int t[],int n) { .....}
/**Moyenne des éléments du tableau **/
float moyenne(int t[],int n) { .....}
/**fonction qui détermine le maximum des éléments du tableau **/
int maximum(int t[],int n) { .....}
/**Une fonction qui ajout un entier x dans le tableau à une position donnée a entrée (int a) au
clavier **/
void ajout(int t[],int n,int x) { .....}
/**Une fonction principale pour tester toutes les fonctions définies à l'aide d'un menu à l'aide
d'une boucle do ...while (choix == - 1 pour quitter le programme, 1 pour la saisie des éléments du
tableau, 2 pour affichage les éléments du tableau, 3 pour le chercher le maximum, 4 pour la moyenne,
5 pour ajouter un élément) ** /

```

Exercice 3 Fusion de deux tableaux triés**Problème:** On dispose de deux tableaux *A* et *B* (de dimensions respectives *N* et *M*), triés par ordre croissant. Fusionner les éléments de *A* et *B* dans un troisième tableau *FUS* trié par ordre croissant.**Méthode:** Utiliser trois indices *IA*, *IB* et *IFUS*. Comparer *A[IA]* et *B[IB]*; remplacer *FUS[IFUS]* par le plus petit des deux éléments; avancer dans le tableau *FUS* et dans le tableau qui a contribué son élément. Lorsque l'un des deux tableaux *A* ou *B* est épuisé, il suffit de recopier les éléments restants de l'autre tableau dans le tableau *FUS*.**NB :** Utiliser le formalisme pointeur à chaque fois que cela est possible.

CORRIGE

Corrigé 1

1) L'exécution de ce programme affichera :

D. a = 1, b = 2

2) Ecrire un programme qui lit un fichier et l'imprime à l'écran, en remplaçant tous les chiffres par le symbole *.

```

/*****/
/** remplacement des chiffres d'un fichier par * ***/
#include <stdio.h>
main()
{
    char c;
    while ((c = getchar()) != EOF)
    {
        if (c >= '0' && c <= '9')
            putchar('*');
        else
            putchar(c);
    }
}
```

```

3) #include <stdio.h>
#include <math.h>
main()
{
    int XA, YA, XB, YB;
    double DIST;
    /* Attention: La chaîne de format que nous utilisons */
    /* s'attend à ce que les données soient séparées par */
    /* une virgule lors de l'entrée. */
    printf("Entrez les coordonnées du point A : XA,YA ");
    scanf("%d,%d", &XA, &YA);
    printf("Entrez les coordonnées du point B : XB,YB ");
    scanf("%d,%d", &XB, &YB);
    DIST=sqrt(pow(XA-XB,2)+pow(YA-YB,2));
    printf("La distance entre A(%d,%d) et B(%d,%d) est %.2f\n",XA, YA, XB, YB, DIST);
    return 0;
}
```

Corrigé 2

```
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
/**Remplissage du tableau**/

void saisie(int t[],int n)
{
    int i;
    for(i=0;i<n;i++){
        printf("les element de T[%d]\n",i);
        scanf("%d",&t[i]);}
}

/**Affichage du tableau**/
void affichage(int t[],int n)
{
    int i;
    printf(" les elements de tableau sont:\n");
    for(i=0;i<n;i++)
        printf("%d ",t[i]);
}

/**Somme des éléments du tableau **/

int somme(int t[],int n)
{
    int i, s=0;
    for(i=0;i<n;i++)
    {
        s=s+t[i];
    }
    return(s);
}

/**Moyenne des éléments du tableau **/

float moyenne(int t[],int n)
{
    int s=somme(t,n);
    float moy=(float)s/n;
    return(moy);
}

/**Maximum des éléments du tableau **/

int maximum(int t[],int n)
{
    int i;
    int max=t[0];

    for(i=0;i<n;i++)
    {
```

```

        if(t[i]>max)
            max=t[i];
    }
    return(max);
}
/**Minimum des éléments du tableau **/

int minimum(int t[],int n)
{
    int i;
    int min=t[0];
    for(i=0;i<n;i++)
    {
        if(t[i]<min)
            min=t[i];
    }
    return(min);
}

/**Nombre de répétitions d'un élément dans le tableau **/

int repetition(int t[],int n,int nbr)
{
    int i,compt=0;

    for(i=0;i<n;i++)
    {
        if(t[i]==nbr)
            compt++;
    }
    if(compt==0)
    {
        printf("l'element n'existe pas");
    }
    return(compt);
}

/** Inversion du tableau **/

void inversion(int t[],int n)
{
    int i;
    for(i=n-1;i>=0;i--)
    {
        printf("%d ",t[i]);
    }
}

/**l'ajout un element dans le tableau**/

void ajout(int t[],int n,int x)
{

```

```

    int i,a;
    printf("donnez la position de la case:\n");
    scanf("%d",&a);
    for(i=n+1;i>a;i--)
    {
        t[i]=t[i-1];
    }
    t[a]=x;
    printf("le tableau apres l'ajout d'un element:\n");
    for(i=0;i<n+1;i++)
    {
        printf("%d ",t[i]);
    }
}

```

/Parité d'un element dans le tableau**/**

```

void parite(int t[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        if(t[i]%2==0)
        {
            printf("%d ",t[i]);
        }
    }
}

```

```

main()
{
    int i,t[50];
    int a,c,x, n;
    printf("taille du tableau <50 \n");
    scanf("%d",&n);
    do{
        printf("\t\t\tFaites votre choix\n\n");
        printf("\t\t\t1- Saisie du tableau\n");
        printf("\t\t\t2- Affichage du tableau\n");
        printf("\t\t\t3- Afficher le maximum du tableau\n");
        printf("\t\t\t4- Afficher le minimum du tableau\n");
        printf("\t\t\t5- Calculer la somme du tableau\n");
        printf("\t\t\t6- Calculer la moyenne du tableau\n");
        printf("\t\t\t7- Le Nombre de repetition d'un element\n");
        printf("\t\t\t8- Inverser le tableau\n");
        printf("\t\t\t9- Ajouter un element\n");
        printf("\t\t\t10- Les nombres paire du tableau \n");
        printf("\t\t\t11- Quitter\n");
        scanf("%d",&c);
        switch(c)
        {
            case 1:saisie(t,n);break;
            case 2:affichage(t,n);break;
            case 3:printf(" le max du tableaux est %d\n\n", maximum(t,n));break;
            case 4:printf(" le min du tableaux est %d", minimum(t,n));break;
            case 5: printf(" la somme du tableaux est %d", somme(t,n));break;
            case 6: printf(" la moyenne du tableaux est %f", moyenne(t,n));break;

```

```

        case 7:printf("entrer l'element a rechercher:\n");
                scanf("%d",&a);
                printf("le nombre de repetition de %d est %d",a,repetition(t,n,a));
                printf("\n\n");break;
        case 8:printf(" le tableaux inverse est:\n");
                inversion(t,n);break;
        case 9:printf("entrer l'element a ajoute:\n");
                scanf("%d",&x);
                ajout(t,n,x);break;
        case 10:printf("les nombres paire sont:\n");
                parite(t,n);break;
    }
}while(c!=11);
}

```

Corrigé 3

```

#include <stdio.h>
main() {
    /* Les tableaux et leurs dimensions */
    int A[50], B[50], FUS[100];
    int N, M;
    int IA, IB, IFUS; /* indices courants */
    /* Saisie des données */
    printf("Dimension du tableau A (max.50) : ");
    scanf("%d", &N );
    printf("Entrer les éléments de A dans l'ordre croissant :\n");
    for (IA=0; IA<N; IA++)
    {
        printf("Elément A[%d] : ", IA);
        scanf("%d", &A[IA]);
    }
    printf("Dimension du tableau B (max.50) : ");
    scanf("%d", &M );
    printf("Entrer les éléments de B dans l'ordre croissant :\n");
    for (IB=0; IB<M; IB++)
    {
        printf("Elément B[%d] : ", IB);
        scanf("%d", &B[IB]);
    }
    /* Affichage des tableaux A et B */
    printf("Tableau A :\n");
    for (IA=0; IA<N; IA++)
        printf("%d ", A[IA]);
    printf("\n");
    printf("Tableau B :\n");
    for (IB=0; IB<M; IB++)
        printf("%d ", B[IB]);
    printf("\n");
    /* Fusion des éléments de A et B dans FUS */
    /* de façon à ce que FUS soit aussi trié. */
    IA=0; IB=0; IFUS=0;
    while ((IA<N) && (IB<M))

```

```

    if(A[IA]<B[IB])
    {
        FUS[IFUS]=A[IA];
        IFUS++;
        IA++;
    }
    else
    {
        FUS[IFUS]=B[IB];
        IFUS++;
        IB++;
    }
/* Si IA ou IB sont arrivés à la fin de leur tableau, */
/* alors copier le reste de l'autre tableau.          */
while (IA<N)
{
    FUS[IFUS]=A[IA];
    IFUS++;
    IA++;
}
while (IB<M)
{
    FUS[IFUS]=B[IB];
    IFUS++;
    IB++;
}
/* Edition du résultat */
printf("Tableau FUS :\n");
for (IFUS=0; IFUS<N+M; IFUS++)
    printf("%d ", FUS[IFUS]);
printf("\n");
return 0;

```