

## TP PROGRAMMATION C: LES FONCTIONS-CORRECTION

### Exercice 1

```

#include<stdio.h>

float moyenne(float, float);

main()
{
    printf("valeur retournee par affiche3= %d\n",affiche3(5));
}

/*****/

void affiche1(){ printf("Bounjour1\n");}

/*****/

void affiche2(int n){
    int i;
    for(i=1;i<=n;i++) printf("bonjour2\n");
}

/*****/

int affiche3(int n){
    int i;
    for(i=1;i<=n;i++) printf("bonjour3\n");
    return 0;
}

```

```

/*****/

```

## Exercice 2

```
#include<stdio.h>

float moyenne(float, float);

main()
{
    float x,y,z,t,M;
    printf("entrer 4 reels: "); scanf("%f%f%f%f",&x,&y,&z,&t);
    M=moyenne(moyenne(x,y),moyenne(z,t));
    printf("la myenne de ces nombres est %f",M);
}

/*****/

float moyenne(float a,float b)
{
    return((a+b)/2.0);
}
```

## Exercice 2

```

#include<stdio.h>
float MIN(float, float);/*prototype de la fonction*/
float MAX(float, float);
main()/*programme principal*/
{ float x,y,z,m,M;
    printf("entrer 3 reels: "); scanf("%f%f%f",&x,&y,&z);
    M=MAX(MAX(x,y),z);
    m=MIN(MIN(x,y),z);
    printf("le MIN est %f le MAX est %f",m,M);
}
/*****/

float MIN(float a, float b){ int minim;
    if(a<b) minim=a;
    else minim=b;
    return minim;}
/*****/

float MAX(float a, float b){ int maxx;
    if(a>b) maxx=a;
    else maxx=b;
    return maxx;
}
/*****/

```

### Exercice 3

```
#include<stdio.h>
#include<math.h>
/*****/
float F(int);
/*****/
main()/*programme principal*/
{ int x;
  printf(" x      F(x) \n");
  printf("-----\n");
  for(x=1;x<=10;x++)
  printf(" %d      %f \n",x,F(x));
  printf("-----\n");
}
/*****/
float F(int y)
{
    return(sin(y)+log(y)-sqrt(y));
}
```

## Exercice 4

```
#include<stdio.h>

/*****/

float convFC(float);

/*****/

main()/*programme principal*/
{ float F;
  printf(" F          C \n");
  printf("-----\n");
  for(F=0;F<=300;F=F+10)
  printf(" %f      %f \n",F,convFC(F));
  printf("-----\n");
}

/*****/

float convFC(float F)
{
    return(5.0/9.0*(F-32));
}
```

## Exercice 5

```
#include<stdio.h>

/*****/

void ndiv(int);

/*****/

main()/*programme principal*/
{
    ndiv(720);
    ndiv(984);
}

/*****/

void ndiv(int n)
{ int compt, i;
  compt=0;
  for(i=1;i<=n;i++)
  {
      if(n%i==0) compt++;
  }
  printf("nombre de diviseur de %d est %d\n",n,compt);
}
```

## Exercice 6:

```
#include<stdio.h>

double fact(int);
double combin(int,int);

/*****/

main()/*programme principal*/
{ printf("C(n=5,p=2)=%lf",combin(5,2)); }

/*****/

double fact(int n)
{ int i; double f;
for(f=i=1;i<=n;i++) f*=i;
return f;
}

double combin(int n,int p)
{return(fact(n)/(fact(p)*fact(n-p))};}
```

## Exercice 7

```
#include<stdio.h>
int puissance(int,int);
/*****/
main()/*programme principal*/
{ int n,m,s,i;
printf("entrer n et m"); scanf("%d%d",&n,&m);
for(s=0,i=1;i<=n;i++) s+=puissance(i,m);
printf("la somme est %d",s);
}
/*****/
int puissance(int x,int n)
{ int i,p;
for(p=i=1;i<=n;i++) p*=x;
return p;
}
```



## TP PROGRAMMATION C: LES TABLEAUX : CORRECTION

### Exercice 1

```
#include<stdio.h>

main()/*programme principal*/
{ int i;
int T[7]={0,0,0,0,0,0,0};
T[0]=0;T[1]=0;T[2]=0;T[3]=0;T[4]=0;T[5]=0;T[6]=0;

for(i=0;i<7;i++)
{
T[i]=0;
}
/* Affichage de T*/
for(i=0;i<7;i++) printf("T[%d]=%d \n",i,T[i]);

}

/*****/

#include<stdio.h>

main()/*programme principal*/
{ int i;
char voy[6]={'a','e','i','o','u','y'};

/* Affichage de T*/

for(i=0;i<6;i++)printf("voy[%d]=%c \n",i,voy[i]);
}
```

## Exercice 2

```
#include<stdio.h>
#define NMAX 50
main()/*programme principal*/
{ int T[NMAX],N,i,S;
printf("entrer la dimension N: (Dim Max est 50)");
scanf("%d",&N);
/*****/
puts("Lecture des elements du tableau:");
for(i=0;i<N;i++){
printf("entrer la valeur de l element T[%d]:?",i);
scanf("%d",&T[i]);
}
/*****/
puts("Affichage des elements du tableau:");
for(i=0;i<N;i++) printf("T[%d]=%d \n",i,T[i]);

for(S=i=0;i<N;i++) S+=T[i];
printf("La somme des elements du tableau est %d",S);
}
```

**Exercice 3:**

```
#include<stdio.h>

const int N=7;

main()/*programme principal*/
{ int T1[N]={2,1,7,4,2,5,7};
  int T2[N]={1,2,1,5,6,4,0};
  int T[N],i;

  for(i=0;i<N;i++) T[i]=T1[i]+T2[i];

  for(i=0;i<N;i++) printf("T[%d]=%d\n",i,T[i]);
}
```

**Exercice 4:**

```
#include<stdio.h>

const int N=3;

main()/*programme principal*/
{ float v1[N]={1,1,0.5};
  float v2[N]={1,0,2};
  int i;
  float s;
  s=0;
  for(i=0;i<N;i++) s+=v1[i]*v2[i];

  printf("Produit scalaire est %f",s);
}
```

## Exercice 5 :

```
#include<stdio.h>

#define N 10

main()/*programme principal*/
{ int T[N]={10,7,9,7,10,6,7,4,8,8};
  int i,j;
  for(i=1;i<N;i++)
    for(j=0;j<i;j++)
      if(T[i]==T[j])T[i]=0;
  puts("Affichage du Tableau: seulement premiere occurrence de chaque element");
  for(i=0;i<N;i++){
    printf("%d \n",T[i]);
  }
  for(j=i=0;i<N;i++)
    if(T[i]!=0){
      T[j]=T[i];
      j++;
    }
  for(i=j;i<N;i++) T[i]=0;

  puts("Affichage du tableau : elemnts restant au debut");
  for(i=0;i<N;i++){
    printf("%d \n",T[i]);
  }
  printf("nombre element different est: %d",j);
}
```

## Exercice 6

```
#include<stdio.h>

#define MAX 50

/*****/

main()/*programme principal*/
{ int Mat[MAX][MAX],N,i,j;

do{ printf("entrer la dimension (<50): "); scanf("%d",&N);
}while(N>MAX || N<=0);
for(i=0;i<N;i++)
    for(j=0;j<N;j++){
        if(i==j)Mat[i][j]=1;
        else Mat[i][j]=0;
    }

puts("Affichage de la matrice unitaire:");
for(i=0;i<N;i++){
    for(j=0;j<N;j++) printf("%d \t",Mat[i][j]);
    printf("\n");
}
}
```

## Exercice 7

```

#include<stdio.h>

#define MAX 50

/*****Declaration de fonctions*/

void lecture(int tab[],int N);
void affiche(int tab[],int N);
int somme(int tab[],int N);

/*****/

main()/*programme principal*/
{ int A[MAX],N,i,S;

do{ printf("entrer la dimension (<50): "); scanf("%d",&N);
}while(N>MAX || N<=0);
puts("Lecture des elements du tableau A:");
lecture(A,N);
puts("Affichage du tableau A:");
affiche(A,N);
S=somme(A,N);
printf("La somme des elments du tabelau est %d",S);
}

/*****Définition des fonctions*****/

void lecture(int tab[],int N)
{ int i;
for(i=0;i<N;i++) {

printf("Entrer element d indice %d: ",i);
scanf("%d",&tab[i]);

}
}

/*****/

```

```
void affiche(int tab[],int N)
{int i;
for(i=0;i<N;i++) printf("tab[%d]=%d\n",i,tab[i]);
}
/*****/

int somme(int tab[],int N)
{int s,i;
for(s=i=0;i<N;i++) s=s+tab[i];
return s;
}
```



**Exercice8 :**

```

/*****/

```

```

int indiceMin(int tab[],int N)

```

```

{

```

```

    int i,Min;

```

```

    Min=0;

```

```

    for(i=1;i<N;i++)

```

```

    {

```

```

        if(tab[i]>tab[Min]) Min=i;

```

```

    }

```

```

    return Min;

```

```

}

```

```

/*****/

```

```

int indiceMax(int tab[],int N)

```

```

{

```

```

    int i,Max;

```

```

    Max=0;

```

```

    for(i=1;i<N;i++)

```

```

    {

```

```

        if(tab[i]<tab[Max]) Max=i;

```

```

    }

```

```

    return Max;

```

```

}

```

```

/*****/

```

```

int somme(int tab[],int N)

```

```

{int s,i;

```

```

for(s=i=0;i<N;i++) s=s+tab[i];

```

```

return s;

```

```

}

```

```

/*****/

```

```

float Moyenne(int tab[],int N)

```

```

    {int s,i;
    for(s=i=0;i<N;i++) s=s+tab[i];
    return (float)s/N;
    }
    /*****/

    int Produitsc(int tab1[],int tab2[],int N)
    {int sc,i;
    for(sc=i=0;i<N;i++) sc=sc+tab1[i]*tab2[i];
    return sc;
    }
    /*****/

    int recherche(int tab[],int N,int v)
    { int i,test=0;
      for(i=0;i<N;i++)
      {
        if(tab[i]==v)
        {
          test=1;
          break;
        }
      }
      return test;
    }
    /*****/
  
```

## TP PROGRAMMATION C: Les types structures et les pointeurs CORRECTION

### Exercice 1

1.

```
#include<stdio.h>

main()
{
int i;
int *p;
i=10;
p=&i;
*p=11;
printf("Valeur de i est %d",i);
}
```

- 2. En utilisant uniquement le « formalisme tableau »,

```
#include<stdio.h>

#define N 10

main()
{ int T[N],i,min;
for(i=0;i<N;i++)
{printf("Entrer element T[%d]: ",i);
scanf("%d",&T[i]);
}
min=T[0];
for(i=1;i<N;i++) if(T[i]<min)min=T[i];
printf("le Plus petit element est: %d",min);
}
```

- 2. En utilisant le « formalisme pointeur », chaque fois que cela est possible

```
#include<stdio.h>
#define N 10
main()
{ int T[N],min;
  int *p;
  for(p=T;p<(T+N);p++)
  {printf("Entrer element T[%d]: ",p-T);
   scanf("%d",&T[p-T]);
  }
  min=T[0];
  for(p=T+1;p<(T+N);p++) if(*p<min)min=*p;
  printf("le Plus petit element est: %d",min);
}
```

## Exercice 2

```
#include<stdio.h>

main()
{ int *p;
  int *q=NULL;
  int v,w;
  v=10;
  p=&v;
  printf("entrer la valeur de v: "); scanf("%d",p);
  w=*p;
  printf("v=%d et w=%d",v,w);
}
```

### Exercice3

```
#include<stdio.h>

void swap(int *a, int *b)
{
    int aux;
    aux=*a;
    *a=*b;
    *b=aux;
}

main()
{ int x,y;
  x=10;
  y=20;
  printf("Avant permutation: x=%d et y=%d\n",x,y);
  swap(&x,&y);
  printf("Après permutation: x=%d et y=%d\n",x,y);
}
```

#### Exercice 4

```
#include<stdio.h>

void incremente(int *i)
{
    *i=*i+1;
}

main()
{ int x,y;
  x=10;
  y=20;
  printf("Avant incrementation: x=%d et y=%d\n",x,y);
  incremente(&x);
  incremente(&y);
  printf("Après incrementation: x=%d et y=%d\n",x,y);
}
```

## Exercice 5

```
#include<stdio.h>
struct complexe{
    float a;
    float b;
};
main()
{ struct complexe c1,c2,s;
printf("entrer partie reelle de c1:");scanf("%f",&c1.a);
printf("entrer partie imaginaire de c1:");scanf("%f",&c1.b);

printf("entrer partie reelle de c2:");scanf("%f",&c2.a);
printf("entrer partie imaginaire de c2:");scanf("%f",&c2.b);

s.a=c1.a+c2.a;
s.b=c1.b+c2.b;

printf("c1=%f + %f i \n c2=%f +%f i \n s=%f +%f i \n",c1.a,c1.b,c2.a,c2.b,s.a,s.b);
}
```



## Exercice 6

```
#include<stdio.h>#define NP 5

struct point{
    float x;
    float y;
};

main()
{ struct point T[NP];
  int i;
  for(i=0;i<NP;i++)
  {
    printf("entrer les coordonnees du point %d:",i);
    scanf("%f%f",&T[i].x,&T[i].y);
  }
  puts("Affichage du tableau de points:");
  for(i=0;i<NP;i++) printf("point %d: (%f,%f)\n",i+1,T[i].x,T[i].y);
}
```