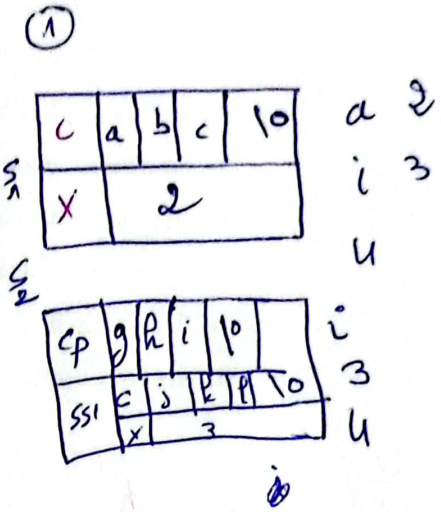
TD3 – Programmation en C

Exercice 1

Donner la représentation des données dans la mémoire et les résultats affichés sur l'écran.

```
void main()
{struct s1
        {char c[4];
         int x;
        };
struct s1 st1={"abc",2};
struct s2 {
        char cp[5];
         struct s1 ss1;
        }st2={"ghi",{"jkl",3}};

        printf("%s%d\n",st1.c[0],st1.x);
        printf("%s%d\n",st2.cp[2],st2.ss1.x);
        st2.ss1.x=st2.ss1.x+1;
        printf("%d",st2.ss1.x);
}
```



Exercice 2: Définir le type de variables et donnez le résultat d'affichage du programme suivant ?

```
#include <stdio.h>
#include <string.h>
int main() {
    voiture v;
    strcpy(v.marque,"Renault");
    voiture w=v;
    strcpy(w.marque,"Kia");
    voiture x=v;

    printf("%s %s %s \n",
v.marque,
       w.marque, x.marque);
```

```
if(w.marque==x.marque){
        strcpy(w.marque,"Ferrari");
        }
    else
        strcpy(x.marque,"Tata");

printf("%s %s %s \n", v.marque,
          w.marque, x.marque);
strcpy(v.marque,"Kia");
if(w.marque!=v.marque){
        strcpy(w.marque,"Cadillac");
        }
    else
        strcpy(x.marque,"Mercedes");
 printf("%s %s %s \n", v.marque,
             w.marque, x.marque);
    return 0;
}
```

Exercice 3

On veut représenter sous forme d'un tableau un agenda téléphonique composé pour chaque enregistrement d'un prénom, nom et numéro de téléphone.

Définir la structure de données en un tableau structuré permettant de représenter l'agenda.

Ecrire un programme en C :

0. qui permet de saisir un certain nombre de personnes dans l'agenda
1. qui permet d'afficher toutes les personnes de l'agenda
2. affichage de tous les n° de téléphone (et noms) correspondant à un prénom donné
3. affichage du n° de téléphone correspondant à un prénom et un nom (unique) donnés (Il n'a donc pas de synonyme)
4. affichage des informations (nom, prénom, tel) correspondant à un rang dans le tableau TABL_AGENDA donnée.
5. modification du n° de téléphone correspondant à un prénom et un nom (unique) donnés (Il n'a donc pas de synonyme)

Correction

```
typedef struct personne{ char nom[20];
                         char prenom[20];
                         char num_tel[20];
                       }; pers;

pers agenda[200];

int n,i;
do{ scanf("%d",&n);
}while(n<=0||n>200);
for(i=0; i<n ; i++)
{ gets(agenda[i].Nom);
  gets(agenda[i].prenom);
  gets(agenda[i].numtel);
}
```

```
#include <string.h>
int i;
char p[20];
gets(p);
for(i=0,i<n,i++)
{ if(strcmp(agenda[i].prenom,p)
  &&strcmp(agenda[i].nom,N)
  { puts(agenda[i].numtel);
    break;}
  else if(i==n)
    printf('n'existe pas');
}
```

```
for(
if(strcmp(agenda[i].nom,N)
{ puts(agenda[i].nom);
  ;
  ;
  break;}
```

```
type def struct Etudiant { int id,
                           struct personne, etud;
                         } etudiant ;
type def struct Personne{
                           char Nom [20];
                           char Personne [20];
                         } pers;

struct Etudiant e;
//init statique
    e.id = 100;
    strcopy(e.etud.Nom, "BenMahmoud");

//init dynamique
    scanf ("%d", Nom.e.id)
     gets(e.etud.Nom);

for(int i=0; i<100 ; i++)
 { scanf ("%.", &Pe[i].id);
   gets (Pe[i].etud.Nom);
```

```
void f( )
{ static int x = 0;
      x ++ ;
   printf ("%d ; x);
}
int main ( )
{ for (int i = 0; i<0 ; i++)
      f( )
   return 0;
}
```

1 1
1 2
1 3
1 4
1 5

Entête

```
float Produit ( float a, float b )
{
   return (a * b)      int, fr
                        Valeur
}
int main ( ) {
   float a = 2.5, b = 4.2;
     produit (a,b)  // passage par Valeur
   return 0;
}
```

```
void permut (int a, int b)
     { int aux;
       aux = a;
       a = b;
       b = aux;
     }
int main ( )
   { int a = 2, b = 3
       Permut (a,b ) ; Printf ("%d %d",a,
   return 0; }
```

V.L

VL

a = 2
b = 3

TD 5 – Programmation C
Les fonctions

Exercice 1 :

Écrire deux fonctions qui calculent la valeur X^N pour une valeur réelle X (type double) et une valeur entière positive N (type int). Elle retourne la valeur X^N comme résultat.

Exercice 2 :

Écrire une fonction MIN et une fonction MAX qui déterminent le minimum et le maximum de deux nombres réels.

Écrire un programme se servant des fonctions MIN et MAX pour déterminer le minimum et le maximum de quatre nombres réels entrés au clavier.

Exercice 3 :

Écrire la fonction NCHIFFRES du type int qui obtient une valeur entière N (positive ou négative) du type long comme paramètre et qui fournit le nombre de chiffres de N comme résultat.

Écrire un petit programme qui teste la fonction NCHIFFRES :

Exemple :

```
Introduire un nombre entier : 6457392
Le nombre 6457392 à 7 chiffres.
```

Exercice 4 :

En mathématiques, on définit la fonction factorielle de la manière suivante :

$0! = 1$

$n! = n*(n-1)*(n-2)* \ldots * 1$ (pour $n>0$)

Écrire une fonction FACT du type double qui reçoit la valeur N (type int) comme paramètre et qui fournit la factorielle de N comme résultat. Écrire un petit programme qui teste la fonction FACT.

TD5_Prog C

Exercice 1 :

```
double pow( double x ; int N)
  {double P = 1 ;  // VL
     for (int i = 0; i < N ; i++){
       Px = x ;}
    return P;}
int main(){  VL
  double P ; int N;
  Printf ("%f", Pow(P,N)); //lecture de P et N
  return 0; }
```

Exercice 2 :

```
float Min (float a, float b)
  { if (a<b)
      return a
    else if (a>b)
      return b
    else return a; }
float Max ( float a , float b)
  { if (a<b)
      return b
      else if( a>b )
        return a
      else return a; }
int main ()
  {
float a, b, c, d ;
 float min , max ;
printf (" donner 4 nombre ", a,b,c,d );
scanf (" %f%f%f%f", a,b,c,d);
```

```
min = Min ( Min (a,b), Min(a,b) );
max = Max( Max(a,b), Max(a,b) ;
printf ("max : %f et min : %f :", min, max );
    return 0; }
```

EXERCICE 3 :

```
void NChiffres (double N){
    int count = 0;
    while ( N != 0){
        count ++;
    N = N / 10      // N /= 10;
    printf ("%d", count); }
```