Les structures

1. Définitions

Structure : Agglomérat hétérogène

Regroupement de types différents sous un même nom. Chaque composant (attribut) est appelé champ et est accessible par son nom (identificateur). L'ensemble est accessible par le nom de la structure.

Exemples

date : regroupement de 3 informations jour, mois et année : 18 mai 2020 ou 18 5 2020

Etudiant :

- Nom : DUPOND
- Prénom : Eric
- Date de naissance : 18 Janvier 1999
- Semestre : IM02
- ….

Définition d'une structure

identificateur

```
struct nom{
  …  //déclarations des champs;
};
```

avec : déclaration des champs

identificateur

```
type1 champ1;
type2 champ2;
……..
typen champn;
```

nom : nom de la structure qui n'est pas une variable

Exemple

```
struct date {
     int jour ;
     int mois;
     int annee ;
     } ;
```

Déclaration des variables basées sur une structure:

d1 et d2 sont des variables basées sur des structures date

```
struct date d1, d2 ;
struct date *d3 ; pointeur de structure
```

Autres façons de définir et de déclarer :

d3 est de type date * et contiendra l'adresse d'une structure date

d3 = &d1 ;

```
struct { …
```

```
} d1, d2;
```
d1 et d2 sont des structures, la structure n'a pas de nom

```
struct date { ……
} d1, d2 ;
```
possible mais préférable de ne pas utiliser.

<u>Initialisation de la structure à la déclaration :</u>

```
typedef struct {
      int jour ;
      char mois[MAX] ;
      int année,
} date ;

date d1 = {18, "mai", 2020} ;
```

<u>Les types des champs sont quelconques :</u>

- un champ peut être une structure
- un champ peut être un tableau

=> on peut imbriquer les types :
structure de structures
structure dont un champ est un tableau.

Remarques les composantes d'un tableau peuvent être des structures
Donc on peut avoir un tableau de structure dont un champ est un tableau et un autre est une structure etc…

2. Création de type basé sur une structure

Création d'un type structure : utilisation du mot réservé `typedef`

```
typedef struct {
……
} nom_du_type;

nom_du_type s1,s2 ;
```

exemple :

```
typedef struct {
      int secondes ;
      int minutes ;
      int heures ;
}horaire ;

horaire h1, h2 ;
```

3. Operateur sur les structures

Seul opérateur défini pour les structures : Affectation

Affectation totale possible :
d1 = d2 => copie de tous les champs de d2 dans d1

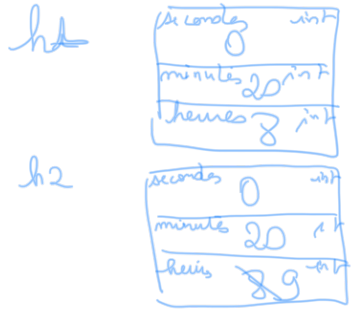
4. Accès aux champs

2 accès possibles selon la nature de l'identificateur :

1) Si c'est une variable, l'accès s'effectue par le nom de la variable dont le type est basé sur une structure suivi d'un point suivi du nom du champ

nom_var_structure.nom_du_champ

```
ex :
horaire h1, h2 ;
h1.secondes = 0 ;
h1.minutes = 20 ;
h1.heures = 8 ;
```



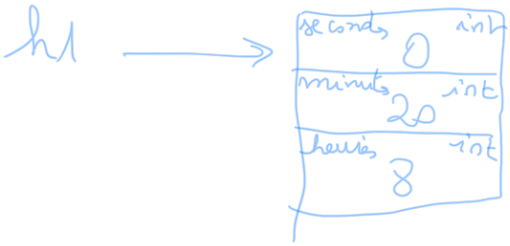
2) Si c'est un pointeur de structure, l'accès s'effectue par le nom du pointeur suivi de -> (tiret et supérieur) suivi du nom du champ

nom_pointeur_structure->nom_du_champ

```
horaire *h1 ;
h1->secondes = 0 ;
h1->minutes = 20 ;
h1->heures = 8 ;
```

equivalent à :

```
(*h1).secondes = 0 ;
(*h1).minutes = 20 ;
(*h1).heures = 8 ;
```

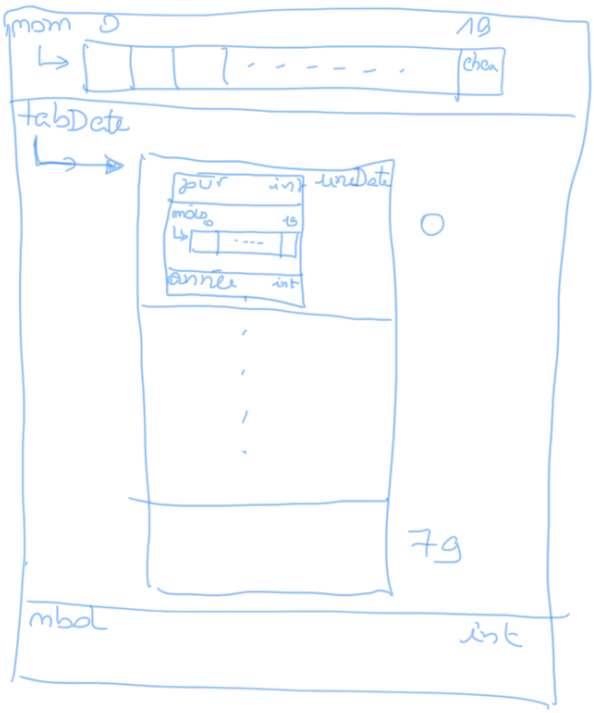



```
#define MAX 80

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

typedf struct {
        char nom[MAX] ;
        date tabDate[MAX],
        int nbr ;
}lesDates;
```

```
lesDates s1 ;
```

Accès à l'année de la dernière date de s1

```
s1.tabDate[s1.nbd - 1].annee
```



s1.nbd - 1

indice de la dernière date du tableau.

indique le nbre de dates dans le tableau

s1 . tabDate [s1 . nbd - 1] . annee

lesDates

lesDates

int

int

uneDate *

uneDate

int

1er lettre du nom de s1

s1 . nom [0]

lesDates

char *

char

dernière lettre du nom de s1

s1.nom [strlen (s1 . nom) - 1]

lesDates

char *

lesDates

char *

int

int

char

5. Passage de structures en paramètres

Comme pour les variables de type simple, on peut passer une structure par valeur ou par adresse selon le rôle du paramètre (donnée, donnée modifiée, résultat).
Une fonction peut retourner une structure mais on favorisera un passage par adresse si le résultat de la fonction est une structure.

```
#define MAX 80

typedef struct {
     int jour ;
     char mois [MAX] ;
     int annee ;
} uneDate ;

void saisieUneDate(uneDate *d)

uneDate saisieUneDate()

void afficheUneDate(uneDate d)

void unAnApres(uneDate *d)
```

```
uneDate saisieUneDate()

void afficheUneDate(uneDate d)

void unAnApres(uneDate *d)
```

```
#include <stdio.h>
#define MAX 12
#define MAXD 20


typedef struct {
    int jour;
    char mois[MAX];
    int annee;
} uneDate;

typedef struct {
    char nom[MAX] ;
    uneDate tabDate[MAX];
    int nbr ;
}lesDates;


void afficheUneDate(uneDate);
void saisieUneDate1(uneDate *);
uneDate saisieUneDate2();
void unAnApres(uneDate *);
void saisieLesDates(lesDates *);
void afficheLesDates(lesDates);


int main()
{
    lesDates annivFamille;
    uneDate D1,D2;


    saisieUneDate1(&D1);
    D2=saisieUneDate2();
    afficheUneDate(D1);
    afficheUneDate(D2);
    unAnApres(&D1);
    afficheUneDate(D1);
    D2=D1;
    afficheUneDate(D2);

    saisieLesDates(&annivFamille);
    afficheLesDates(annivFamille);
    return 0;


}
```

```c
void afficheUneDate(uneDate d)
{
    printf("\nla date est   : %d %s %d\n", d.jour, d.mois, d.annee);
}

void saisieUneDate1(uneDate *d)
{
    printf("\nentrez le jour svp :");
    scanf("%d", &d->jour);
    fflush(stdin);
    printf("\nentrez le mois svp :");
    scanf("%s", d->mois); // mois est un pointeur sur le premier
caractère
    printf("\nentrez l'anne svp :");
    scanf("%d", &d->annee);
}

uneDate saisieUneDate2()
{
    uneDate d;  //resultat
    printf("\nentrez le jour svp :");
    scanf("%d", &d.jour);
    fflush(stdin);
    printf("\nentrez le mois svp :");
    scanf("%s", d.mois); // mois est un pointeur sur le premier
caractère
    printf("\nentrez l'anne svp :");
    scanf("%d", &d.annee);
    return d;
}

void unAnApres(uneDate *d)
{
    d->annee++;
}
void saisieLesDates(lesDates *anniv)
{
    int i; // parcours du tableau de dates
    fflush(stdin);
    printf("donner le nom de la structure de données comportant
plusieurs dates");
    scanf("%s", anniv->nom);
    printf("\ncombien de dates voulez vous (<=%d)",MAXD);
    scanf("%d", &anniv->nbr );
    for (i=0;i<anniv->nbr;i++)
        saisieUneDate1( &anniv->tabDate[i]);
        // anniv->tabDate[i] = saisieUneDate2( anniv->tabDate[i]);

}

void afficheLesDates(lesDates anniv)
{   int i; // parcours du tableau de dates
    printf("\n nom : %s", anniv.nom);
    for(i=0; i<anniv.nbr; i++)
        afficheUneDate(anniv.tabDate[i]);
}
```