

# Cours Complexité des Algorithmes

Introduction

Prof. A. BOUZIDI

# Prérequis

- Algorithmiques
- Langage C/C++

# Objectifs

- Complexité Algorithmique
- Programmation itératif/réursive
  - Algorithmes de Tris
  - Algorithmes de Recherche
- Structure de données dynamique
  - Piles, Files, Listes
- Résolution des problèmes NP-difficile
  - Exemple : Problème voyageur de commerce

# Organisation du module

- Cours
- TP
- Evaluation:
  - TP noté
  - Projet à réaliser
  - Contrôle à la fin

# Plan

- Rappel:
- Recherche et Tri
- Récursivité
- Complexité Algorithmique
- Structure de données
  - Piles
  - Files
  - Listes
  - Graphes

# Rappel

## Pointeurs

### Exemple

```
void main()
```

```
{
```

```
    int x = 10 , y;
```

```
    int *p;
```

```
    p = &x;
```

```
    y = *p;
```

```
    *p = 20;
```

```
}
```

Déclaration d'un pointeur

Affecter adresse de la variable x

Affecter la valeur de la variable dont l'adresse est p

???

# Rappel

## Pointeurs

```
void main()
{
    int A = 1, B = 2, C = 3;
    int *P1, *P2;
    P1 = &A;
    P2 = &C;
    *P1 = (*P2)++;
    P1 = P2;
    P2 = &B;
    *P1 -= *P2;
    ++ *P2;
    *P1 *= *P2;
    A = ++*P2**P1;
    P1 = &A;
    *P2 = *P1 /= *P2;
}
```

```
// P1 pointe sur A
// P2 pointe sur C
// A = 3 C = 4
// P1 pointe sur C
// P2 pointe sur B
// C = 2
// B = 3
// C = 6
// A = 24
// P1 pointe sur A
// B = 6
```

# Rappel

## *Pointeurs et tableaux*

### *Exemple*

```
int A[10], B;
```

```
int *P ;
```

```
P = A ;
```

```
B = *(P+1);
```

```
B = *(P+2);
```

```
...
```

```
B= *(P+i);
```

```
// P pointe sur A[0], P = &A[0];
```

```
//P+1 pointe sur le contenu de A[1]
```

```
// P+2 pointe sur le contenu de A[2]
```

```
...
```

```
// P+i pointe sur le contenu de A[i]
```



# Rappel

## Pointeurs : Allocation dynamique

Supposons que nous avons besoin d'un bloc de mémoire pour une image de niveaux de gris (0..255) de 1024X1024 unsigned char.

```
unsigned char *img;
```

```
img = malloc(1024*1024) ;
```

```
free(img);
```

```
int *p = malloc(sizeof(int));
```

```
free(p);
```

```
double *pT = malloc(sizeof(double) * 10);
```

```
free(pT);
```

# Les structures

# Enoncé

1. Ecrire une structure de type `Personne` {nom, prenom, age, metier}
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes  
`{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
5. Initialisez la variable `Etudiant` avec les valeurs suivantes  
`{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

# Solution

- 1.Ecrire une structure de type `Personne` {nom, prenom, age, metier}
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}
- 5.Initialisez la variable `Etudiant` avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P\_Etudiant}
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

# Solution

1. Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
5. Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

## Rappel :

### Structures

- Équivalent de l'enregistrement en algorithmique
- Type hétérogène composé de plusieurs *champs* pouvant avoir chacun leur propre type
- Syntaxe :

```
struct nom_type {  
    type1 champ1 ;  
    type2 champ2 ;  
    type3 champ3 ;  
    ...  
};
```
- déclaration : `struct nom_type objet;`
- La taille réservée en mémoire est la somme des tailles des champs

### Structures

- Ou :

```
struct nom_type {  
    type1 champ1 ;  
    type2 champ2 ;  
    type3 champ3 ;  
    ...  
} objet ;
```
- Accès à différents membres :  
`objet.champi` le ième membre de l'objet

### Structures

- Utilisation de `typedef` :

```
typedef struct etudiant Etudiant ;  
...  
Etudiant e ;
```

# Solution

1. Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
5. Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

Rappel :

## Pointeurs sur des structures

- Comme pour les types de base, on peut utiliser des « pointeurs sur structure » :  
    `Etudiant e, *pe ;`  
    `pe = &e ;`  
    ⇒ `pe` contiendra l'adresse du premier champ de la structure
- Utilisation du symbole `->` pour accéder aux champs *dans le cas des pointeurs* :  
    `pe->CNE = 12 ;`  
    Ou  
    `(*pe).CNE = 12 ;`

# Solution

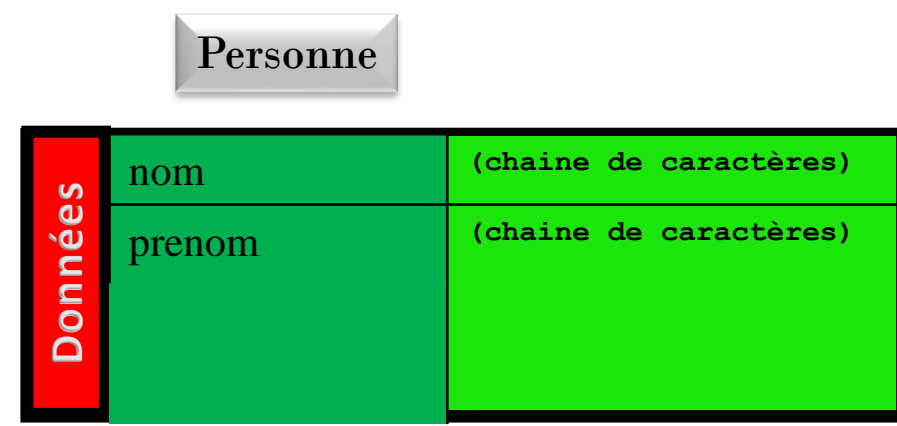
1. Ecrire une structure de type **Personne** {nom, prenom, age, metier}
2. Créez la variable **Etudiant** de type **Personne**.
3. Insérez dans la structure **Personne** un pointeur **pere** de type **Personne**.
4. Créez la variable **P\_Etudiant**. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}
5. Initialisez la variable **Etudiant** avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P\_Etudiant}
6. Affichez les informations concernant **Etudiant** ainsi que celles concernant son père.

```
struct Personne {
};
```

# Solution

- 1.Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
- 5.Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

```
struct Personne {  
    char nom[20];  
    char prenom[20];  
};
```





# Solution

- 1.Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
- 5.Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

```
struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
};
```

|          |        |                        |
|----------|--------|------------------------|
| Personne |        |                        |
| Données  | nom    | (chaîne de caractères) |
|          | prenom | (chaîne de caractères) |
|          | age    | (entier)               |

# Solution

- 1.Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
- 5.Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

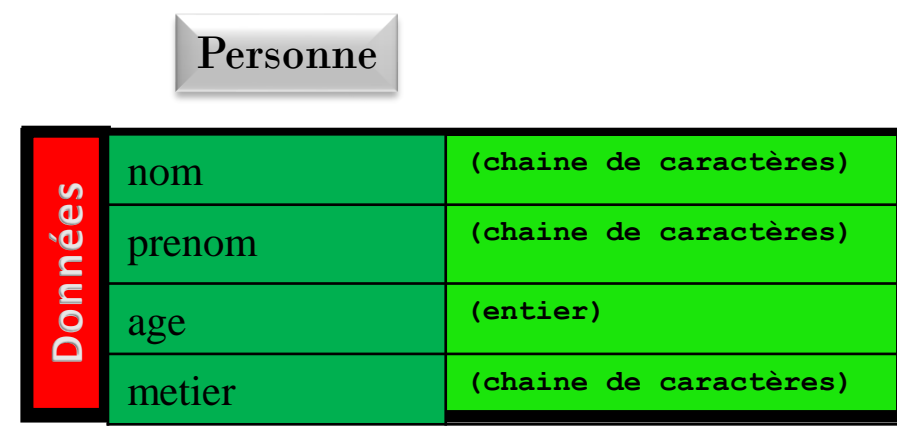
```
struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
    char metier[20];  
};
```

|          |        |                        |
|----------|--------|------------------------|
| Personne |        |                        |
| Données  | nom    | (chaîne de caractères) |
|          | prenom | (chaîne de caractères) |
|          | age    | (entier)               |
|          | metier | (chaîne de caractères) |

# Solution

- 1.Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
- 5.Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

```
typedef struct Personne {  
    char nom[20];  
    char prenom;  
    int age;  
    char metier[20];  
};
```



# Solution

```
typedef struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
    char metier[20];  
};
```

1.Ecrire une structure de type **Personne** {nom, prenom, age, metier}

2. Créez la variable **Etudiant** de type **Personne**.

3. Insérez dans la structure **Personne** un pointeur **pere** de type **Personne**.

4. Créez la variable **P\_Etudiant**. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}

5.Initialisez la variable **Etudiant** avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P\_Etudiant}

6. Affichez les informations concernant **Etudiant** ainsi que celles concernant son père.

# Solution

```
typedef struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
    char metier[20];  
};
```

1.Ecrire une structure de type Personne {nom, prenom, age, metier}

2. Créez la variable **Etudiant** de type **Personne**.

3. Insérez dans la structure Personne un pointeur pere de type Personne.

4. Créez la variable P\_Etudiant. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}

5.Initialisez la variable Etudiant avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P\_Etudiant}

6. Affichez les informations concernant Etudiant ainsi que celles concernant son père.

## Personne Etudiant;

# Solution

```
typedef struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
    char metier[20];  
};
```

1. Ecrire une structure de type `Personne` {nom, prenom, age, metier}

2. Créez la variable `Etudiant` de type `Personne`.

3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.

4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}

5. Initialisez la variable `Etudiant` avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = `P_Etudiant`}

6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

```
typedef struct Personne {  
  
    char nom[20];  
  
    char prenom[20];  
  
    int age;  
  
    char metier[20];  
  
};
```

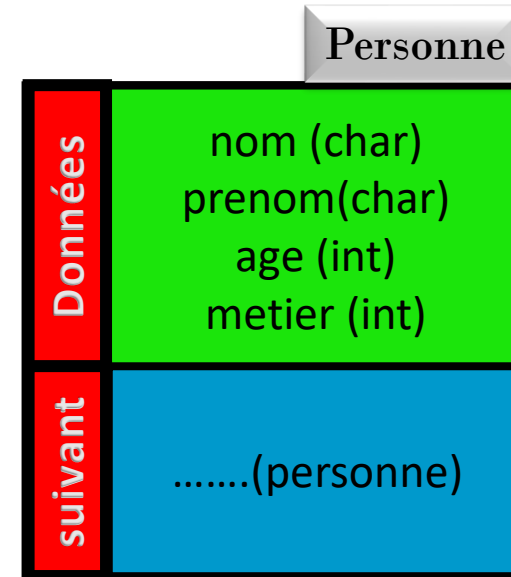
| Données | nom (char)<br>prenom(char)<br>age (int)<br>metier (int) |
|---------|---|
|         |   |

# Solution

```
typedef struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
    char metier[20];  
};
```

1. Ecrire une structure de type `Personne` {nom, prenom, age, metier}
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}
5. Initialisez la variable `Etudiant` avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = `P_Etudiant`}
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

```
typedef struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
    char metier[20];  
    Personne .....;  
};
```

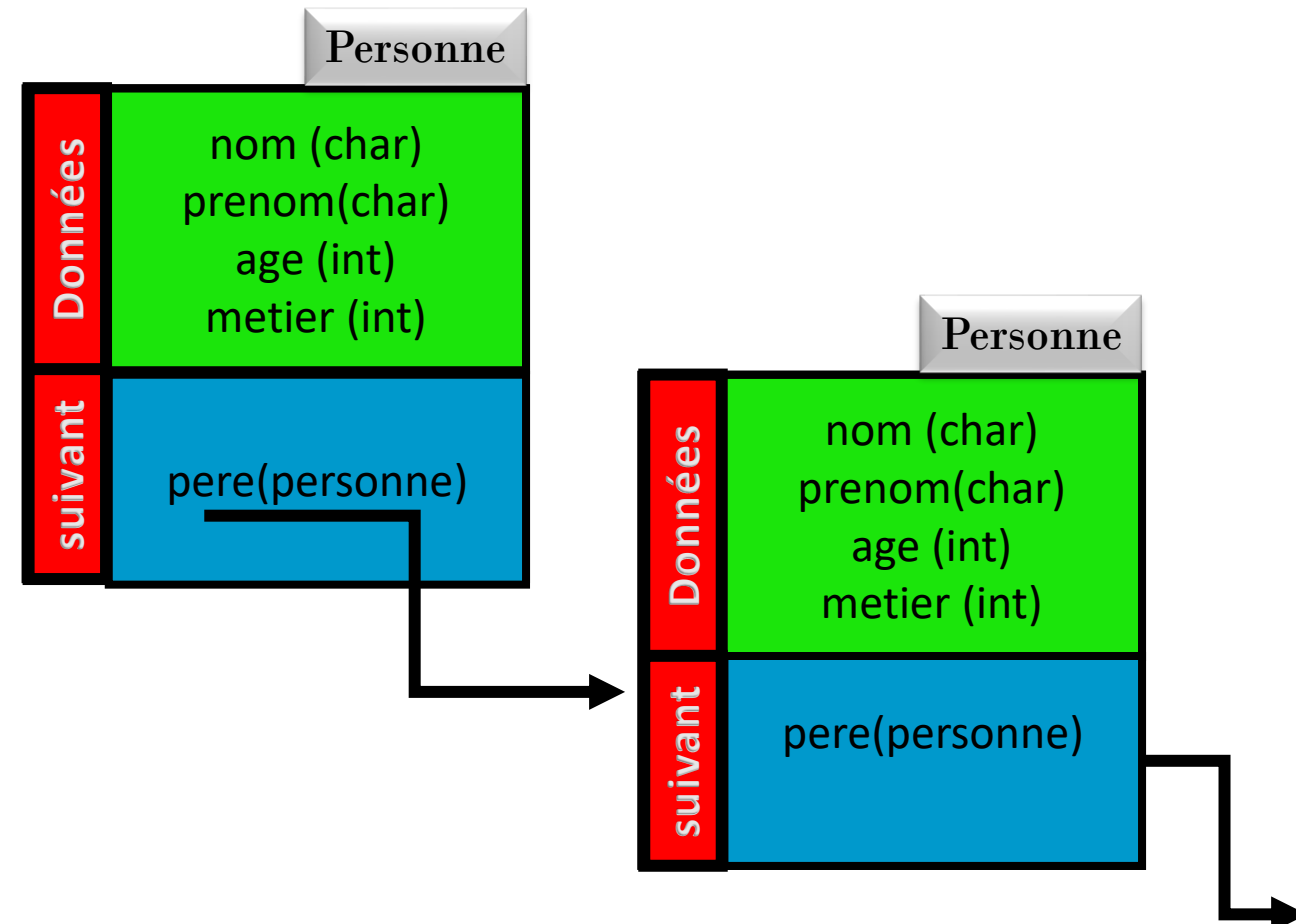


# Solution

```
typedef struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
    char metier[20];  
    Personne *pere;  
};
```

1. Ecrire une structure de type `Personne` {nom, prenom, age, metier}
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}
5. Initialisez la variable `Etudiant` avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P\_Etudiant}
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

```
typedef struct Personne {  
    char nom[20];  
    char prenom[20];  
    int age;  
    char metier[20];  
    Personne *pere;  
};
```





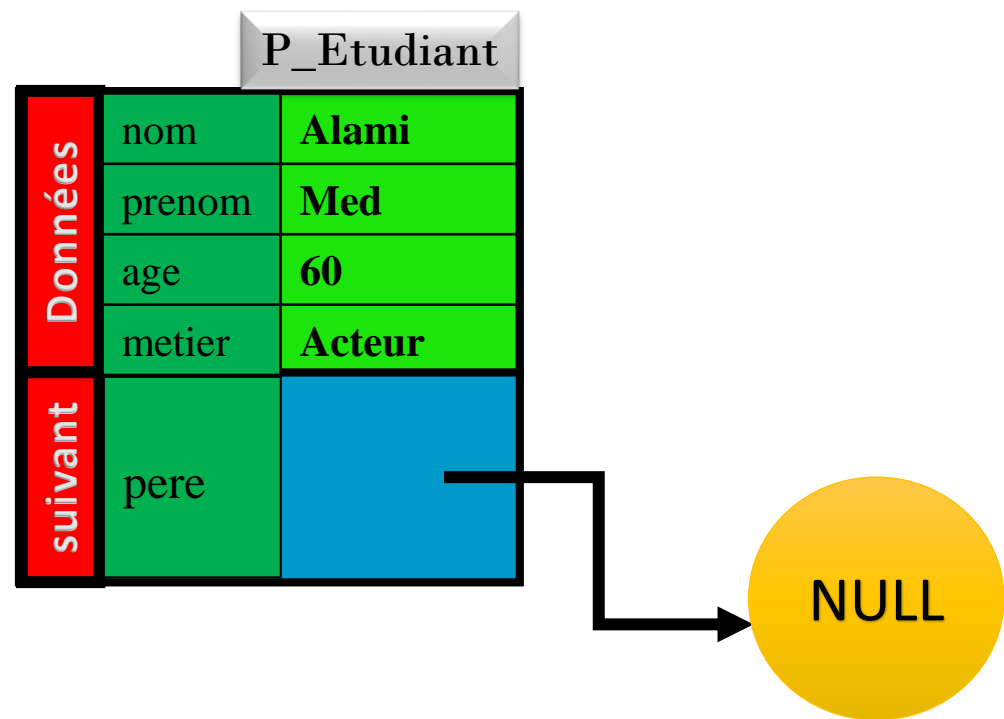
# Solution

- 1.Ecrire une structure de type `Personne` {nom, prenom, age, metier}
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}
- 5.Initialisez la variable `Etudiant` avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P\_Etudiant}
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

# Solution

- 1.Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
- 5.Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

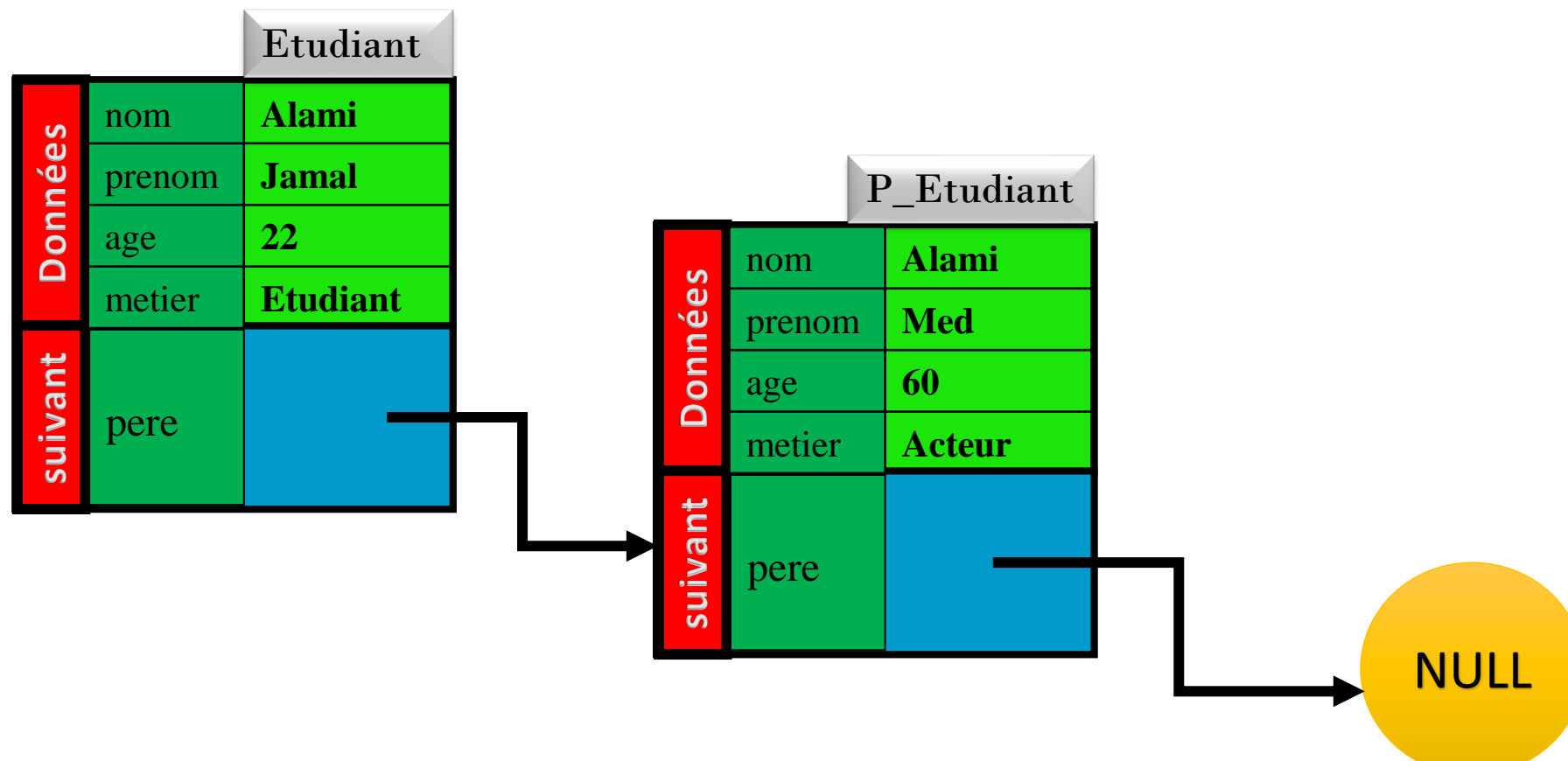
Personne P\_Etudiant = { "Alami", "Med", 60, "Acteur", NULL };



# Solution

1. Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
5. Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

`Etudiant = { "Alami", "Jamal", 22, "Etudiant", &P_Etudiant };`

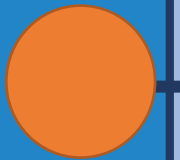



# Solution


1. Ecrire une structure de type `Personne {nom, prenom, age, metier}`
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes `{nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}`
5. Initialisez la variable `Etudiant` avec les valeurs suivantes `{nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P_Etudiant}`
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

```
printf(" les informations de l'etudiant %s , %s  sont : ", ?????????, ?????????);  
printf("\t Age : %d \t metier : %d \n", ?????????, ?????????);  
printf("\n Nom du pere : %s \t prenom du pere : %s \n", ?????????, ?????????);  
printf("\t metier : %s \t Age : %d \n \n", ?????????, ?????????);
```

**Etudiant.Nom\_champs**

| Etudiant |        |   |
|----------|--------|---|
| Données  | nom    | Alami   |
|          | prenom | Jamal   |
|          | age    | 22  |
|          | metier | Etudiant  |
| suivant  | pere   |  |

| P_Etudiant |        |   |
|------------|--------|---|
| Données    | nom    | Alami   |
|            | prenom | Med   |
|            | age    | 60  |
|            | metier | Acteur  |
| suivant    | pere   |  |



NULL

# Solution

1. Ecrire une structure de type `Personne` {nom, prenom, age, metier}
2. Créez la variable `Etudiant` de type `Personne`.
3. Insérez dans la structure `Personne` un pointeur `pere` de type `Personne`.
4. Créez la variable `P_Etudiant`. Et initialisez là avec les données suivantes {nom = Alami, prenom = Med, age = 60, metier = Acteur, pere = NULL}
5. Initialisez la variable `Etudiant` avec les valeurs suivantes {nom = Alami, prenom = Jamal, age = 22, metier = Etudiant, pere = P\_Etudiant}
6. Affichez les informations concernant `Etudiant` ainsi que celles concernant son père.

```
#include<stdio.h>
#include<stdlib.h>
using namespace std;

typedef struct Personne {
    char nom[20];
    char prenom[20];
    int age;
    char metier[20];
    struct Personne* pere;
};

void main(){
    Personne Etudiant;
    Personne P_Etudiant = { "Alami", "Med", 60, "Acteur", NULL };
    Etudiant = { "Alami", "Jamal", 22, "Etudiant", &P_Etudiant };
    printf(" les informations de l'etudiant %s , %s sont : ", Etudiant.nom, Etudiant.prenom);
    printf("\t Age : %d \t metier : %d \n", Etudiant.age, Etudiant.metier);
    //info du pere
    printf("\n Nom du pere : %s \t prenom du pere : %s \n", Etudiant.pere->nom, Etudiant.pere->prenom);
    printf("\t metier : %s \t Age : %d \n \n", Etudiant.pere->metier, Etudiant.pere->age);
    printf(" \n ");

    system("pause");
}
```

