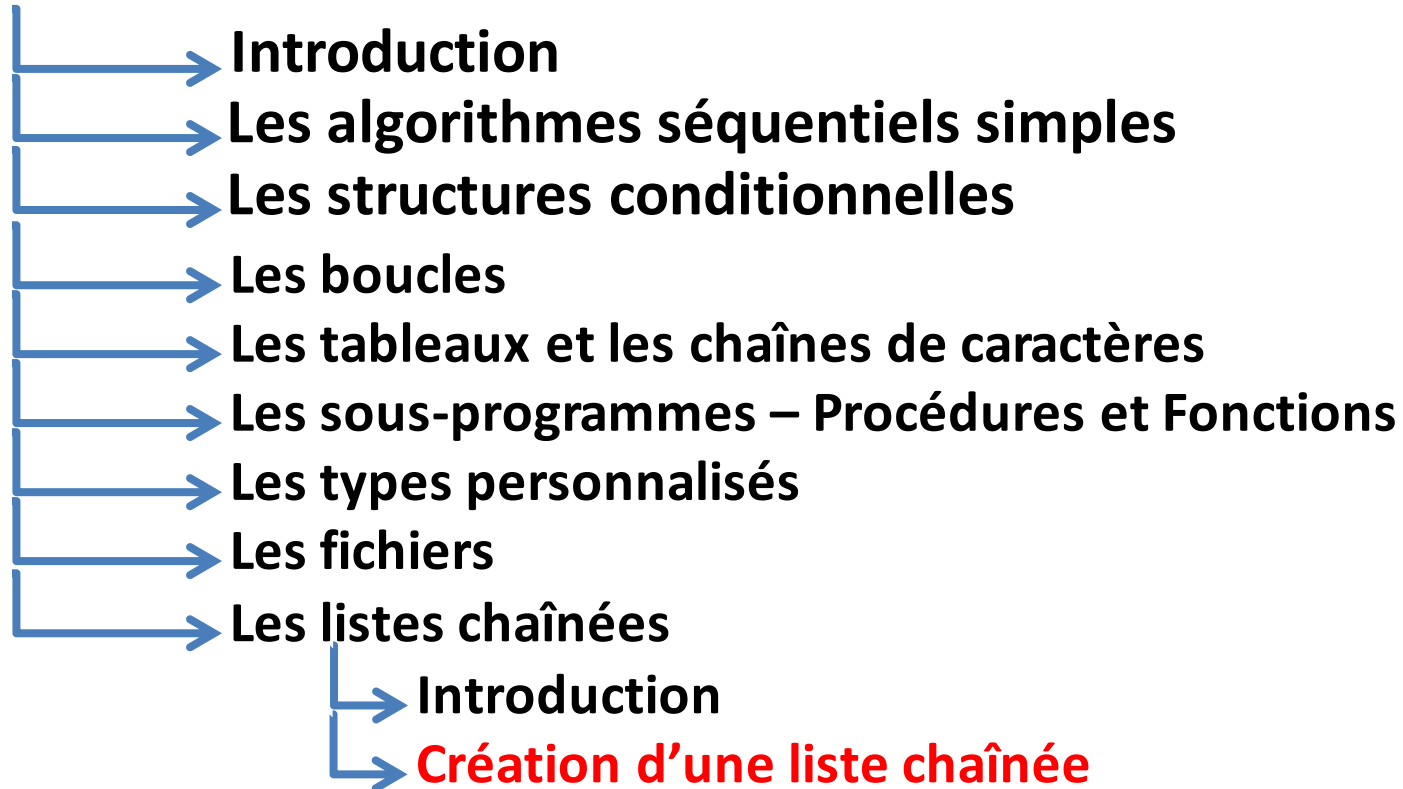


Algorithmique et structure de données



Djelloul BOUCHIHA
bouchiha.dj@gmail.com
2020-2021

Liste chaînée

- **Définition :**

- ☞ un ensemble d'éléments de même type.
- ☞ chaque élément est lié à son successeur par un pointeur.

- **Structure d'un élément :**

- ☞ les données à stocker et un pointeur vers le prochain élément (successeur) de la liste.

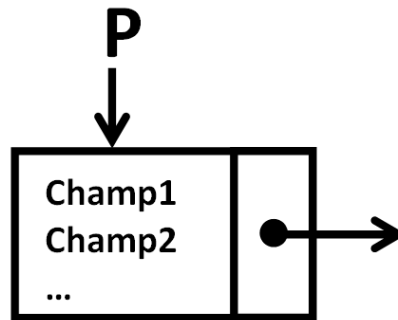
```
typedef struct element
{
    <Type_données1>  Champ1 ;
    <Type_données2>  Champ2 ;
    ...
    struct element  *Suivant;
} Element ;
```

Liste chaînée

- Déclaration d'un pointeur vers élément :

Element *P ; // L'adresse de l'élément est rangée dans la variable P

- Représentation d'un pointeur vers élément :



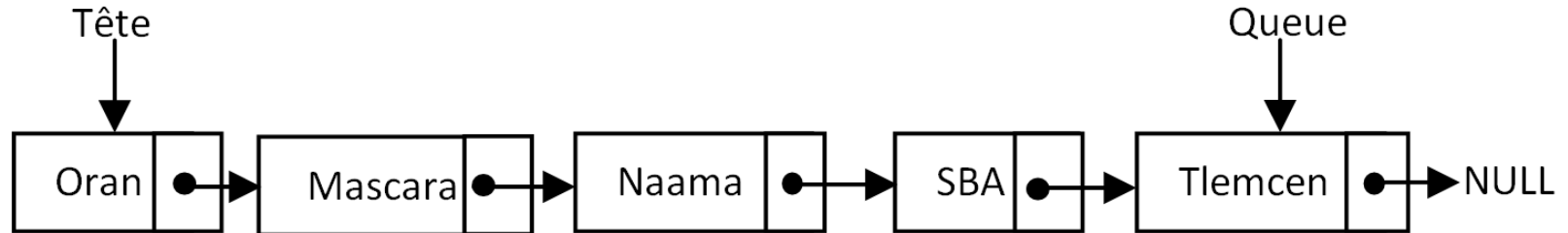
- Accès :

☞ Accès aux champs : **P->Champ1**, **P->Champ2**...

☞ Accès à l'élément suivant : **P->Suivant**.

Création d'une liste chaînée

- Liste des villes à créer :



- Définir la structure Ville :

```
typedef struct ville
{
    char Des[20] ;
    struct ville *Suivant ;
} Ville;
```

- Déclarer Tete et Queue :

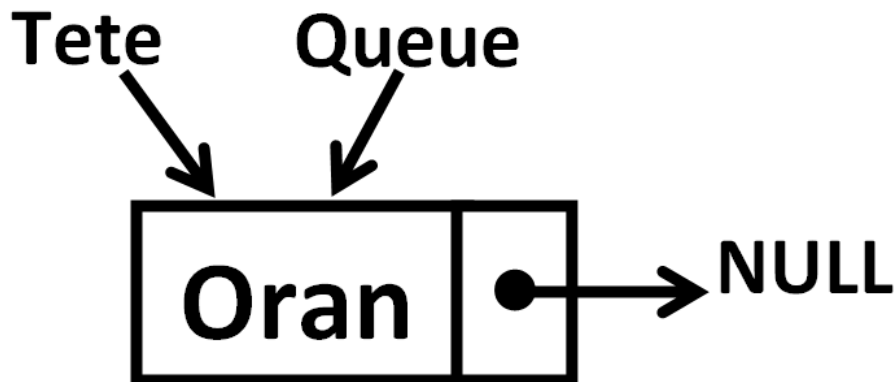
```
Ville *Tete, *Queue ;
```

Création d'une liste chaînée

- Créer le premier élément :

```
Tete = (Ville*)malloc(sizeof(Ville)) ;  
strcpy(Tete->Des, "Oran") ;  
Queue = Tete ;  
Queue->Suivant = NULL ;
```

- Représentation du premier élément :

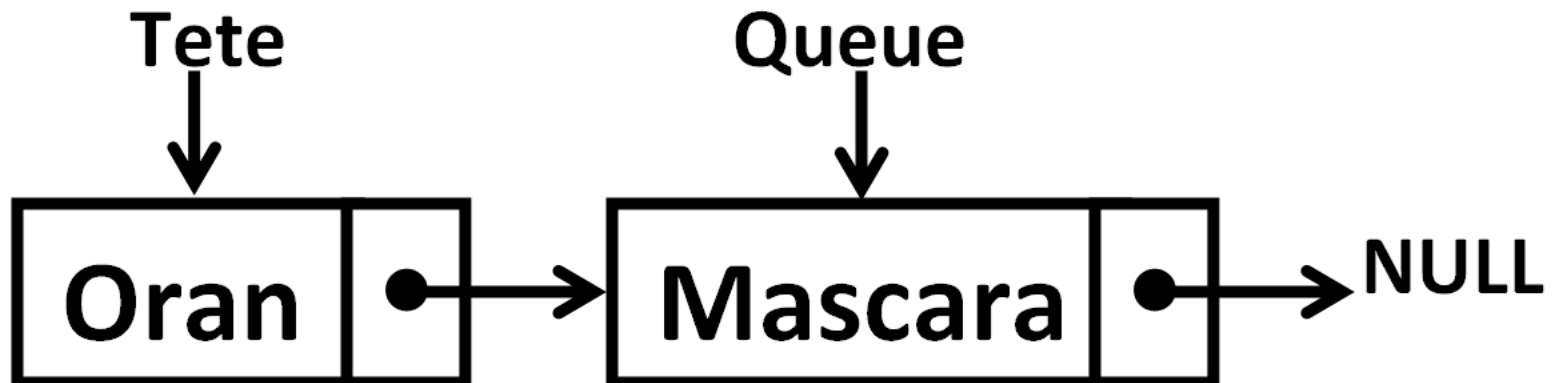


Création d'une liste chaînée

- Ajouter le deuxième élément :

```
Queue->suivant = (Ville*)malloc(sizeof(Ville));  
Queue = Queue->Suivant ;  
strcpy(Queue->Des, "Mascara") ;  
Queue->Suivant = NULL ;
```

- Représentation de la liste :



Création d'une liste chaînée

- Procédure d'ajout d'un élément :

```
void Ajouter_Q (char V[20])
{
    Queue->suivant = (Ville*)malloc(sizeof(Ville));
    Queue = Queue->Suivant ;
    strcpy(Queue->Des, V) ;
    Queue->Suivant = NULL ;
}
```

Création d'une liste chaînée

- Programme de création de la liste :

```
#include<stdio.h>
#include<string.h>
#include <stdlib.h>
typedef struct ville
{
    char Des[20] ;
    struct ville *Suivant ;
} Ville;

Ville *Tete, *Queue ;

/* Procédure d'ajout d'un élément */
void Ajouter_Q (char V[20])
{
    Queue->Suivant = (Ville*)malloc(sizeof(Ville));
    Queue = Queue->Suivant ;
    strcpy(Queue->Des, V) ;
    Queue->Suivant = NULL ;
}

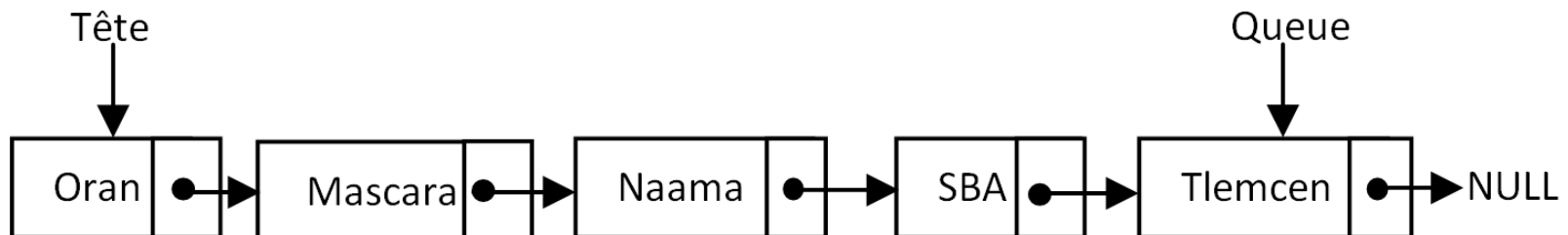
main()
{
    /* Créer le premier élément de la liste */
    Tete = (Ville*)malloc(sizeof(Ville)) ;
    strcpy(Tete->Des, "Oran") ;
    Queue = Tete ;
    Queue->Suivant = NULL ;

    /* Ajouter le reste des éléments */
    Ajouter_Q("Mascara") ;
    Ajouter_Q("Naama") ;
    Ajouter_Q("SBA") ;
    Ajouter_Q("Tlemcen") ;
}
```


Affichage d'une liste chaînée

- Procédure d'affichage des éléments de la liste :

```
/* Procédure d'affichage des éléments de la liste */  
void Afficher_Liste()  
{  
    Ville *P ;  
    P = Tete;  
    while (P != NULL) {  
        printf("%s\n", P->Des) ;  
        P = P->Suivant;  
    }  
}
```



Merci