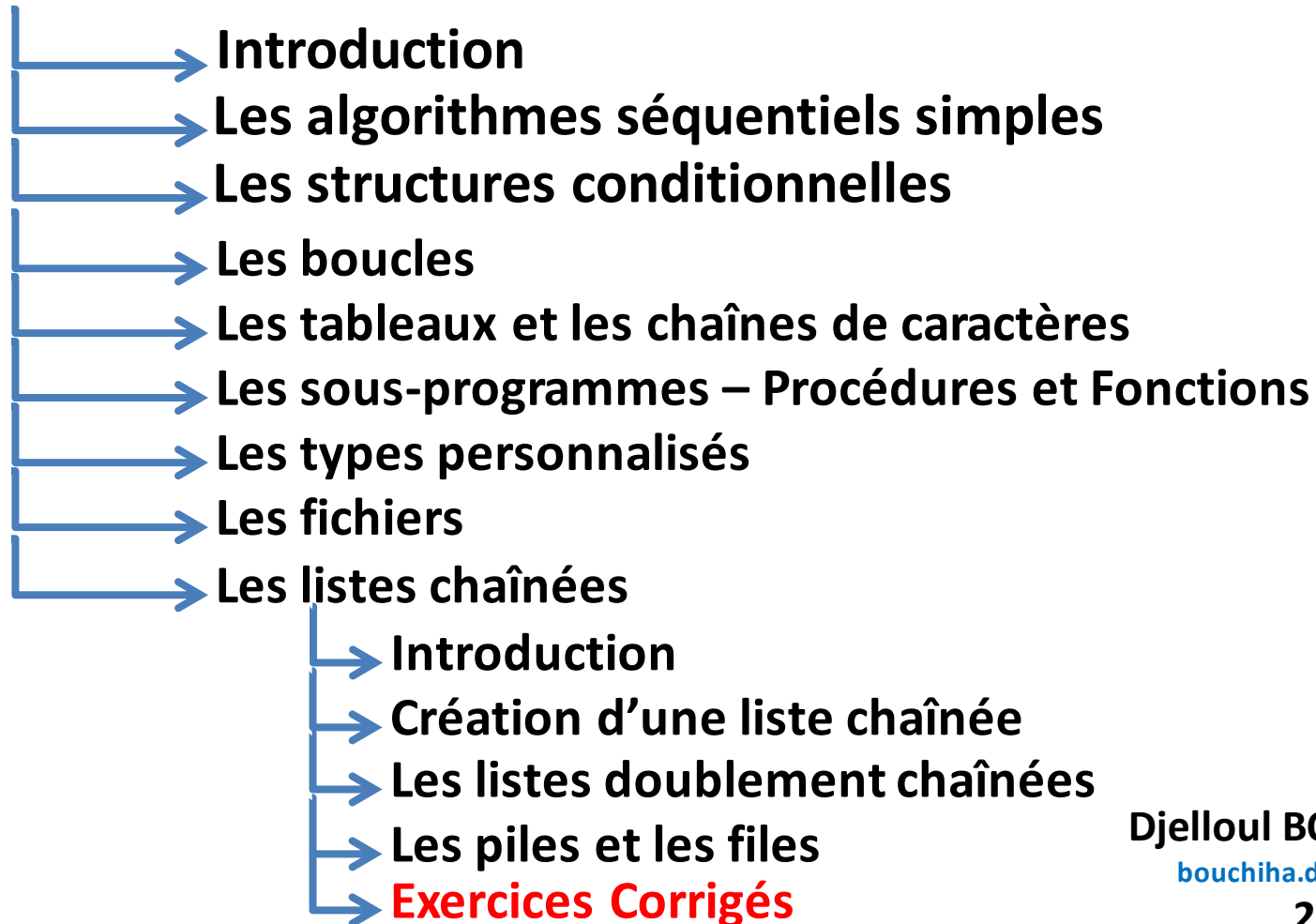


Algorithmique et structure de données



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

Exercices

- **Exercice 1 :**

- ☞ Ecrire en C une procédure permettant d'ajouter un élément au début (en Tête) de la liste simplement chaînée créée dans le cours. La procédure possède comme paramètre la ville à ajouter. Elle possède l'entête : **void Ajouter_T (char V[20])**

- **Exercice 2 :**

- ☞ Ecrire en C une procédure permettant de supprimer la tête de la liste doublement chaînée créée dans le cours, et récupérer l'espace mémoire occupé par cet élément. La procédure possède l'entête : **void Supprimer_T()**

- **Exercice 3 :**

- ☞ Ecrire en C une fonction entière permettant la recherche d'une ville dans la liste chaînée créée dans le cours. La fonction retourne 1 si l'élément existe et 0 si l'élément n'existe pas. La fonction possède comme paramètres la ville à rechercher. Elle possède l'entête : **int Rechercher(char V[20])**

Corrigés

- Solution 1 :

```
/* Procédure d'ajout d'un élément au début de la liste */  
void Ajouter_T(char V[20])  
{  
    Ville *P;  
    P = (Ville*)malloc(sizeof(Ville)) ;  
    strcpy(P->Des, V) ;  
    P->Suivant = Tete ;  
    Tete = P;  
    if (Queue == NULL) Queue = Tete ;  
}
```

Corrigés

- **Solution 2 :**

/* Procédure de suppression de la tête de la liste */

void Supprimer_T()

{

if (Tete != NULL)

if (Tete->Suivant == NULL) {

free(Tete) ;

Tete = NULL ;

Queue = NULL ;

}

else {

Tete = Tete->Suivant ;

free(Tete->Precedent) ;

Tete->Precedent = NULL ;

}

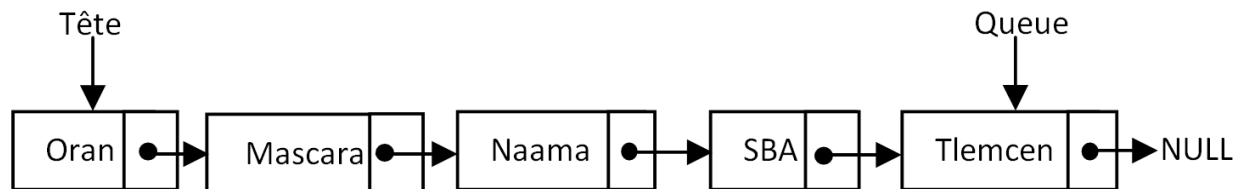
}

Corrigés

- Solution 3 :

/* Fonction de recherche d'un élément dans la liste */

```
int Rechercher(char V[20])  
{  
    Ville *P ;  
    int i ;  
    P = Tete;  
    i = 0;  
    while ((P != NULL) && (! i))  
        if ( ! strcmp(P->Des, V)) i = 1;  
        else P = P->Suivant ;  
  
    return i ;  
}
```



Merci