



Université des Sciences et de la Technologie Houari Boumédiène

Faculté d'Informatique
Département Informatique

Module : Algorithme et structure de données

Rapport du projet de TP "Livres électroniques"

Réalisé par :

Zakaria REGUIEG 212137033656
Ikram KERRIS 212131054231

Année universitaire : 2022 / 2023

1-Introduction:

1.1-représentation du mini projet:

Consiste à élaborer une solution pour permettre au auteurs d'organiser leur livre électronique ou il peuvent ajouter ou supprimer des chapitres ,modifier le contenu d'un chapitre donné, afficher les chapitres et le nombre de pages du livre .

1.2 la méthodologie :

- les structures de données utilisé :
- liste simplement chaînée, les enregistrements, les vecteurs.
- emploi des deux méthodes itératives et récursif(2 fct rec).
- la complexité des fonctions : $O(1)$, $O(n)$, $O(n^2)$.
- environnement de développement: Codeblocks.
- Bibliothèque utilisé:
<stdio.h>, <stdlib.h>, <string.h>, <stdbool.h>, <windows.h>

2-partie théorique: (le fonctionnement):

À l'exécution du programme, il s'affiche un tableau de bord représentant le menu qui contient toutes les fonctionnalités qu'il peut l'auteur interpréter .

```
-----menu-----
1- Ajouter un chapitre
2-afficher les chapitres selon id
3-afficher le nombre de pages
4-modifier le contenu d'un chapitre
5-afficher le contenu du plus grand chapitre
6-afficher le contenu du plus petit chapitre
7-supprimer un chapitre
8-affiche selon nombre de pages
9-clear screen
10-exit

veuillez choisir une fonction :
```

2.1-Ajouter un chapitre:

au choix de cette fonctionnalité s'exécute la procédure **'inser'** complexité **O(n)** qui prend en entrée un pointeur vers la tête de la liste et une position, elle insère un chapitre à cette position dans la liste. Si la position est 1 le chapitre est inséré en tête de liste. Sinon, la fonction parcourt la liste jusqu'à arriver à la position spécifiée, et insère le chapitre à la position donnée (milieu ou fin).

Si le nouveau chapitre est inséré en fin de livre, son titre du chapitre suivant est défini sur le titre du premier chapitre du livre, Si le nouveau chapitre n'est pas inséré

en fin de liste, le titre du chapitre suivant est défini sur le titre du chapitre qui le suit dans la liste.

Elle prend en considération le changement de l'Id des chapitres qui sont après le nouveau chapitre.

Cette procédure utilise une autre fonction de contrôle qui s'appelle '**check**' qui permet de vérifier si la position spécifiée est valide pour l'insertion ou non . Elle compte le nombre de chapitres dans la liste et vérifie si la position est comprise entre 1 et le nombre de chapitres + 1. Si c'est le cas, la fonction renvoie true, sinon elle renvoie false.

2.2-Afficher la liste selon l'id:

En exécutant la fonction '**aff**' **complexité $O(n)$** qui affiche les chapitres en ordre croissant selon leur Id en parcourant la liste .

Si la liste est vide, la fonction affiche un message d'erreur indiquant que la liste est vide.

2.3-Affiche le nombre de pages :

Elle affiche le nombre de pages du livre en utilisant la fonction réursive '**nbpages**' **complexité $O(n)$** qui prend en entrée la tête de la liste et elle fait la somme

des nombres de pages des chapitres en parcourant la liste d'une manière récursive.

2.4-Modifier le contenu d'un chapitre :

en exécutant la fonction '**modifcontent**' de **complexité $O(n)$** qui prend en entrée deux arguments :

t : un pointeur qui représente la tete de la liste

idch : un entier, qui représente l'identifiant du chapitre dont on souhaite modifier son contenu.

La fonction commence par demander à l'utilisateur de saisir le nouveau contenu qu'il souhaite attribuer à l'élément de la liste chaînée, en utilisant la fonction scanf.(s'il ya un espace dans le contenu à saisir le programme va ignorer les prochaines instructions de saisie le même problème pour la fonction gets et fgets)

la solution: indiquer au utilisateur qu'il faut utiliser le tiret de huit “_” a la place de l'espace “ ” .

Ensuite, la fonction parcourt la liste chaînée en utilisant une boucle while, jusqu'à trouver le chapitre de la liste dont l'identifiant (Id) correspond à celui passé en argument (idch).

Une fois l'élément trouvé, le nouveau contenu sera copier dans le champs (contenu) du chapitre sélectionné.

2.5-Afficher les informations du plus grand chapitre

En exécutant l'Ap '**maxchapitre**' de **complexité $O(n)$** prend en entrée la tête de la liste.

Si la tete =NULL alors elle affiche un message d'erreur

Sinon la procédure commence par initialiser un pointeur temp à la valeur de tete, et initialiser une variable entière max à la valeur du nombres de pages du premier chapitre.

Ensuite, la fonction parcourt la liste chaînée en utilisant une boucle while, et à chaque itération, elle compare la valeur du champ 'nbpages' du chapitre courant à la valeur de max. Si la valeur de nbpages est supérieure à max, alors max prend la valeur de nbpages.

Une fois que la boucle a parcouru tous les éléments de la liste, max contient la valeur du champ nbpages de l'élément de la liste ayant le nombre de pages maximum.

La fonction initialise à nouveau temp à la valeur de tete, et parcourt la liste chaînée jusqu'à trouver le chapitre ayant le champ nbpages égal à la valeur de max.

Une fois le chapitre trouvé, la fonction affiche toutes les informations du chapitre en utilisant les fonctions printf et puts.

2.6-Afficher les informations du plus petit chapitre :

L'ap '**minchap**' de complexité **$O(n)$** utilise d'abord la fonction récursive '**f_m**' pour trouver l'élément de la liste chaînée ayant le nombre de pages minimum. Si la liste est vide , la fonction affiche un message indiquant que la liste est vide. Sinon, la fonction initialise un élément de type chap aux informations du chapitre pointé par t, puis appelle la fonction '**f_m**' en passant l'élément suivant de la liste (pointé par t->suiv) et l'adresse de l'élément mini en argument.

Une fois que f_m a trouvé le chapitre ayant minimum, la fonction '**minchap**' affiche toutes les informations de cet élément en utilisant les fonctions printf et puts.

2.7-Supprimer un chapitre :

La fonction '**supprimer**' **complexité $O(n^2)$** prend en entrée deux arguments :

t : pointeur sur le début de la liste.

pos : un entier, qui représente la position du chapitre que l'on souhaite supprimer.

La fonction commence par vérifier si l'élément de la liste chaînée à la position '**pos**' existe, en appelant la fonction '**checksup**', Si l'élément n'existe pas, la

fonction demande à l'utilisateur de saisir une nouvelle position et recommence à partir de l'étiquette debut2. la fonction vérifie si la liste est vide, Si c'est le cas, la fonction affiche un message indiquant que la liste est vide.

Sinon, si la position de l'élément à supprimer est égale à 1, cela signifie que l'élément à supprimer est le premier de la liste. Dans ce cas, une insertion au début, puis met à jour les 'Id' des chapitres suivants en parcourant la liste chaînée à l'aide d'une boucle while.

Enfin, la fonction met à jour le 'titre du chapitre suivant' du dernier chapitre en lui affectant le titre du nouveau premier chapitre.

Si la position de l'élément à supprimer est supérieure à 1, cela signifie que l'élément à supprimer se trouve au milieu ou à la fin de la liste. Dans ce cas, la fonction utilise deux pointeurs p et q pour parcourir la liste chaînée jusqu'à l'élément à supprimer, en utilisant une boucle for. Une fois que les pointeurs p et q sont positionnés sur l'élément à supprimer et sur l'élément précédent, la fonction met à jour le pointeur de l'élément précédent pour qu'il pointe vers l'élément suivant de l'élément à supprimer, en utilisant

$q \rightarrow \text{suiv} = p \rightarrow \text{suiv}$. Si l'élément à supprimer est le dernier de la liste, la fonction met à jour le titre du chapitre suivant de l'avant-dernier élément de la liste en utilisant

`strcpy((q->info).titrechapsuiv,((*t)->info).titrechap)`.

Sinon, la fonction met à jour le titre du chapitre suivant de l'avant-dernier élément de la liste en utilisant

`strcpy((q->info).titrechapsuiv,((p->suiv)->info).titrechap)`. Enfin, la fonction met à jour les identifiants des éléments suivants de la liste en parcourant la liste.

2.7-Afficher selon le nombre (ordre croissant) :

La fonction '**supprimer**' de **complexité $O(n^2)$** commence par demander à l'utilisateur de saisir le nombre de chapitres à afficher ,La fonction déclare un tableau de chap de taille n, un élément de type chap temporaire tmp, et deux variables d'itération i et j.

La fonction parcourt la liste chaînée en utilisant un pointeur P et stocke chaque élément de la liste dans le tableau tab, elle trié le tableau en ordre croissant selon le nombre des pages,une fois le tableau tab trié la fonction affiche les informations des chapitres d'une manière croissante selon le nombre de pages.

2.7-Clear screen:

Elle permet de réinitialiser l'affichage du console en utilisant la fonction 'system("cls")' de la bibliothèque <windows.h> pour une meilleure expérience utilisateur.

3-L'exécution:

3.1-L'ajout des chapitres:

```
entre la position du chapitre : 1
-----Important: il faut r
entrez le titre : chapitre1
entrez le nombres de pages : 39
entrez le contenu : bonjour_c'est_le_chapitre_1
10-Sortir          1-Afficher le menu:
```

ajout du 2eme chapitre

```
entre la position du chapitre : 2
-----Important: il faut remplacer l'
entrez le titre : chapitre2
entrez le nombres de pages : 290
entrez le contenu : bounjour_mes_freres_cest_le_chapitre_2
10-Sortir          1-Afficher le menu:
```

3.2-Affichage selon Id:

```
veuillez choisir une fonction : 2
id = 1
le titre: chapitre1
nbpages = 39
le contenu: bonjour_c'est_le_chapitre_1
le titre du chapsuiv: chapitre2

id = 2
le titre: chapitre2
nbpages = 290
```

3.3 Afficher le nombre de pages :

```
veuillez choisir une fonction : 3  
le nombre de pages est: 329  
10-Sortir          1-Afficher le menu:
```

3.4 modifier le contenu :

```
veuillez choisir une fonction : 4  
entrez la position du chapitres a modifier 1  
entrez le nouveau contenu  
je_mesuis_trompe_oups1
```

Affichage apres la modification du chapitre 1

```
le titre: chapitre1  
nbpages = 39  
le contenu: je_mesuis_trompe_oups1  
le titre du chapsuiv: chapitre2
```

3.5 afficher le plus grand chapitre:

```
veuillez choisir une fonction : 5  
les informations du plus grand chapitre:  
id = 2  
le titre: chapitre2  
le nombre de pages: nbpages = 290  
le contenu: bounjour_mes_freres_cest_le_chapitre_2  
le titre du chapitre suivant: chapitre1  
10-Sortir          1-Afficher le menu:
```

3.6 afficher le plus petit chapitre:

```
veuillez choisir une fonction : 6
les informations du plus petit chapitre:
id = 0
le titre: chapitre1
le nombre de pages: nbpages = 39
le contenu: je_mesuis_trompe_oups1
le titre du chapitre suivant: chapitre2
10-Sortir          1-Afficher le menu:
```

3.7- Supprimer un chapitre:

```
veuillez choisir une fonction : 7
entrez la position du chapitre a supprimer: 1
10-Sortir          1-Afficher le menu:
```

Affichage apres la suppression du premier chap

```
veuillez choisir une fonction : 2
id = 1
le titre: chapitre2
nbpages = 290
le contenu: bonjour_mes_freres_cest_le_chapitre_2
le titre du chapitre suiv: chapitre2
```

3.8- affiche selon nombre de pages:

```
veuillez choisir une fonction : 8
Donner le nombre de chapitres : 2
1 - le titre chapitre2
nbpages = 290
le contenu: bonjour_mes_freres_cest_le_chapitre_2
2 - le titre nouveau_chapitre_ajoute_pour_le_YO
nbpages = 345
le contenu: ok
10-Sortir          1-Afficher le menu:
```