



**Réalisé par : - Ikram Oussarhan**  
**- Fatiha Ourahou**





# Introduction

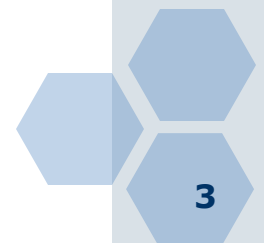
**Les bibliothèques jouent un rôle crucial dans la promotion de la culture, de l'éducation et de la lecture. Avec l'avènement des technologies numériques, la gestion efficace des bibliothèques est devenue essentielle pour répondre aux besoins croissants des utilisateurs et pour optimiser l'organisation interne. Ce projet vise à développer un système de gestion de bibliothèque utilisant des structures de données telles que les listes, piles, files et arbres, ainsi que des fichiers pour assurer la persistance des données.**





# Plan

- 1 **Structure de données**
- 2 **Fonctionnalités**
- 3 **Présentation du code**
- 4 **Exécution du code**
- 5 **Conclusion**





## ❖ **Structure de données :**

**Le programme est composé de plusieurs structures de données :**

- Livre : contient les informations d'un livre (titre, auteur, prix, ISBN).**
- Emprunteur : contient les informations d'un emprunteur (nom, prénom, adresse e-mail).**
- Liste d'emprunteurs : permet de gérer la liste des emprunteurs.**
- Pile de livres : permet de gérer la pile des livres.**





## ❖ **Fonctionnalités :**

**Le programme offre plusieurs fonctionnalités pour interagir avec la bibliothèque :**

### **Gestion des emprunteurs :**

- Ajouter un emprunteur.
- Afficher la liste des emprunteurs.
- Supprimer un emprunteur.
- Compter le nombre d'emprunteurs.

### **Gestion des livres :**

- Ajouter un livre.
- Afficher la liste des livres disponibles.
- Afficher les livres d'un auteur spécifique.
- Compter le nombre de livres.
- Emprunter un livre.
- Rendre un livre.



# Présentation du Code

```
File Edit Search View Project Execute Tools AStyle Window Help
[Icons]
(globals)
Project Classes Debug main.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  typedef struct {
6      char titre[50];
7      char auteur[50];
8      int prix;
9      int ISBN;
10     struct Livre *next;
11 } Livre;
12
13 typedef struct {
14     char nom[20];
15     char prenom[20];
16     char adresse_email[50];
17     struct emprunteurs *next;
18 } emprunteurs;
19
20 typedef struct {
21     int taille;
22     emprunteurs *debut;
23     emprunteurs *fin;
24 } List_emprunteur;
25
26 typedef struct {
27     Livre *sommet;
28     int taille;
29 } Pile_Livres;
30
```



# Code (suite)

main.c

```
31 void ajouter_emprunteurs(List_emprunteur *L, char N[20], char PN[20], char AE[50]) {
32     emprunteurs *Nouvel_emprunt = (emprunteurs *)malloc(sizeof(emprunteurs));
33     strcpy(Nouvel_emprunt->nom, N);
34     strcpy(Nouvel_emprunt->prenom, PN);
35     strcpy(Nouvel_emprunt->adresse_email, AE);
36     Nouvel_emprunt->next = NULL;
37     if (L->debut == NULL) {
38         L->debut = Nouvel_emprunt;
39         L->fin = Nouvel_emprunt;
40     } else {
41         L->fin->next = Nouvel_emprunt;
42         L->fin = Nouvel_emprunt;
43     }
44     L->taille++;
45     printf("L'emprunteur %s %s a été ajouté avec succès.\n", N, PN);
46 }
47
48 void supprimer_emprunteur(List_emprunteur *L, char nom[20], char prenom[20]) {
49     emprunteurs *courant = L->debut;
50     emprunteurs *precedent = NULL;
51
52     while (courant != NULL) {
53         if (strcmp(courant->nom, nom) == 0 && strcmp(courant->prenom, prenom) == 0) {
54             if (precedent == NULL) {
55                 L->debut = courant->next;
56             } else {
57                 precedent->next = courant->next;
58             }
59             free(courant);
60             L->taille--;
61             return;
62         }
63         precedent = courant;
64         courant = courant->next;
65     }
66     printf("L'emprunteur %s %s n'a pas été trouvé.\n", nom, prenom);
67 }
68
```



# Code (suite)

```
main.c
69 void afficherListeEmprunteurs(List_emprunteur *L) {
70     if (L->taille <= 1) {
71         printf("La liste des emprunteurs est vide.\n");
72     } else {
73         emprunteurs *courant = L->debut;
74         while (courant != NULL) {
75             printf("Nom : %s\n", courant->nom);
76             printf("Prénom : %s\n", courant->prenom);
77             printf("Adresse e-mail : %s\n", courant->adresse_email);
78             courant = courant->next;
79         }
80     }
81 }
82
83 void ajouter_livre(Pile_Livres *pile, char titre[20], char auteur[20], int prix, int ISBN) {
84     Livre *Nouvel_Livre;
85     Nouvel_Livre = (Livre *)malloc(sizeof(Livre));
86     strcpy(Nouvel_Livre->titre, titre);
87     strcpy(Nouvel_Livre->auteur, auteur);
88     Nouvel_Livre->prix = prix;
89     Nouvel_Livre->ISBN = ISBN;
90     Nouvel_Livre->next = pile->sommet;
91     pile->sommet = Nouvel_Livre;
92     pile->taille++;
93 }
94
95 void supprimer_livre(Pile_Livres *pile, char titre[20]) {
96     Livre *temps = NULL;
97     Livre *courant = pile->sommet;
98     Livre *P = NULL;
99
100     if (strcmp(courant->titre, titre) == 0) {
101         pile->sommet = courant->next;
102         free(courant);
103         pile->taille--;
104         return;
105     }
106 }
```





# Code (suite)

```
main.c
107 while (courant != NULL && courant->next != NULL) {
108     P = courant->next;
109     if (strcmp(P->titre, titre) == 0) {
110         temps = courant->next;
111         courant->next = P->next;
112         free(temps);
113         pile->taille--;
114         return;
115     }
116     courant = courant->next;
117 }
118 }
119
120 void afficherListeLivres(Pile_Livres *pile) {
121     if (pile->taille <= 1) {
122         printf("La pile est vide.\n");
123     } else {
124         Livre *courant = pile->sommet;
125         while (courant != NULL) {
126             printf("Titre : %s\n", courant->titre);
127             printf("Auteur : %s\n", courant->auteur);
128             printf("Prix : %d\n", courant->prix);
129             printf("ISBN : %d\n", courant->ISBN);
130             courant = courant->next;
131         }
132     }
133 }
134
135 void afficherLivresParAuteur(Pile_Livres *pile, char auteur[20]) {
136     int found = 0;
137     Livre *courant = pile->sommet;
138 }
```



# Code (suite)

```
main.c
139 while (courant != NULL) {
140     if (strcmp(courant->auteur, auteur) == 0) {
141         found = 1;
142         printf("Titre : %s\n", courant->titre);
143         printf("Auteur : %s\n", courant->auteur);
144         printf("Prix : %d\n", courant->prix);
145         printf("ISBN : %d\n", courant->ISBN);
146         printf("\n");
147     }
148     courant = courant->next;
149 }
150
151 if (!found) {
152     printf("L'auteur %s n'a pas d'œuvres répertoriées.\n", auteur);
153 }
154 }
155
156 void emprunter_livre(Pile_Livres *pile, char titre[20]) {
157     Livre *courant = pile->sommet;
158
159     while (courant != NULL) {
160         if (strcmp(courant->titre, titre) == 0) {
161             printf("Le livre %s a été emprunté.\n", titre);
162             supprimer_livre(pile, titre);
163             return;
164         }
165         courant = courant->next;
166     }
167     printf("Le livre %s n'a pas été trouvé.\n", titre);
168 }
169
170 void rendre_livre(Pile_Livres *pile, char titre[20]) {
171     Livre *Nouvel_Livre;
172     Nouvel_Livre = (Livre *)malloc(sizeof(Livre));
173     strcpy(Nouvel_Livre->titre, titre);
174     Nouvel_Livre->next = pile->sommet;
175     pile->sommet = Nouvel_Livre;
176     pile->taille++;
177 }
178
```



# Code (suite : main)

main.c

```
179 int main() {
180     char auteur[20], titre[20], nom[20], prenom[20], adresse_email[50];
181     int choix = 0, choix1 = 0, choix2 = 0, n = 0, prix = 0, ISBN = 0;
182     List_emprunteur list = {0, NULL, NULL};
183     Pile_Livres pile = {NULL, 0};
184
185     do {
186         printf("*****\n");
187         printf("Taper 1 pour lister les paramètres d'emprunteurs :\n");
188         printf("Taper 2 pour lister les paramètres des livres :\n");
189         printf("Taper 3 pour quitter\n");
190         printf("*****\n");
191         printf("Saisissez votre choix :");
192         scanf("%d", &choix);
193
194         switch (choix) {
195             case 1 : {
196                 do{
197                     printf("*****\n");
198                     printf("Apuyer sur 1 pour ajouter un emprunteur.\n");
199                     printf("Apuyer sur 2 pour afficher la liste des emprunteurs.\n");
200                     printf("Apuyer sur 3 pour supprimer un emprunteur.\n");
201                     printf("Apuyer sur 4 pour compter le nombre des emprunteurs.\n");
202                     printf("Apuyer 5 pour quitter\n");
203                     printf("*****\n");
204                     printf("Saisissez votre choix :");
205                     scanf("%d", &choix1);
206                     switch (choix1) {
207                         case 1:{
208                             printf("Entrez le nom de l'emprunteur :");
209                             scanf(" %[^\\n]s", nom);
210                             printf("Entrez le prénom de l'emprunteur :");
211                             scanf(" %s", prenom);
212                             printf("Entrez l'adresse e-mail de l'emprunteur :");
213                             scanf(" %s", adresse_email);
214                             ajouter_emprunteurs(&list, nom, prenom, adresse_email);
215                             break;}
216                         case 2:{
217                             printf("Liste des emprunteurs :\n");
218                             afficherListeEmprunteurs(&list);
219                             break;}
```



# Code (suite : main)

main.c

```
220 case 3:{
221     printf("Entrez le nom de l'emprunteur à supprimer :");
222     scanf(" %s", nom);
223     printf("Entrez le prénom de l'emprunteur à supprimer :");
224     scanf(" %s", prenom);
225     supprimer_emprunteur(&list, nom, prenom);
226     break;}
227 case 4:{
228     printf("Le nombre d'emprunteurs est : %d\n", list.taille);
229     break;}
230 }
231 }while(choix1!=5);
232 break; }
233
234 case 2 : {
235     do{
236         printf("*****\n");
237         printf("Apuyer sur 1 pour ajouter un livre.\n");
238         printf("Apuyer sur 2 pour afficher les livres disponibles.\n");
239         printf("Apuyer sur 3 pour afficher les livres d'un auteur.\n");
240         printf("Apuyer sur 4 pour compter le nombre des livres.\n");
241         printf("Apuyer sur 5 pour emprunter un livre.\n");
242         printf("Apuyer sur 6 pour rendre un livre.\n");
243         printf("Apuyer 7 pour quitter\n");
244         printf("*****\n");
245         printf("Saisissez votre choix :");
246         scanf("%d", &choix2);
247         switch (choix2) {
248             case 1:{
249                 printf("Combien de livres vous souhaitez ajouter :");
250                 scanf("%d", &n);
251                 for (int i = 0; i < n; i++) {
252                     printf("*****\n");
253                     printf("Entrer les détails du livre numéro %d : \n", i + 1);
254                     printf("*****\n");
255                     printf("Titre :");
256                     scanf(" %[^\n]s", titre);
257                     printf("Auteur :");
258                     scanf(" %[^\n]s", auteur);
259                     printf("Prix :");
260                     scanf("%d", &prix);
```



# Code

```
main.c
261     printf("ISBN :");
262     scanf("%d", &ISBN);
263     ajouter_livre(&pile, titre, auteur, prix, ISBN);
264 }
265 break;
266 }
267 case 2: {
268     printf("Liste des livres disponibles :\n");
269     afficherListeLivres(&pile);
270     break;}
271 case 3:{
272     printf("Saisissez le nom de l'auteur :");
273     scanf(" %[^\\n]s", auteur);
274     afficherLivresParAuteur(&pile, auteur);
275     break;
276 case 4:
277     printf("Le nombre de livres est : %d\\n", pile.taille);
278     break;}
279 case 5:{
280     printf("Saisissez le titre du livre à emprunter :");
281     scanf(" %[^\\n]s", titre);
282     emprunter_livre(&pile, titre);
283     break;}
284 case 6:{
285     printf("Saisissez le titre du livre à rendre :");
286     scanf(" %[^\\n]s", titre);
287     rendre_livre(&pile, titre);
288     break;}
289 }
290 }while(choix2!=7);
291 break; }
292 }
293 } while (choix != 3);
294 return 0;
295 }
```

**Fin du programme**