Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes - RABAT

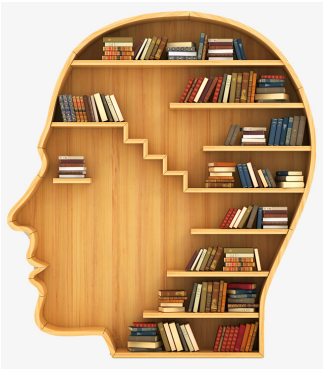
Rapport de Projet de programmation : prêts de livre

Réalisé par :

aicha MOUDAR
houda JOUHAR

Encadré par :

Pr. karima El MOUMANE



Année Scolaire 2020/2021

remerciements

Nous voudrons tout d'abord adresser toute notre gratitude à notre professeur karima MOUMANE, pour sa confiance, sa disponibilité et surtout cette opportunité pour bien maîtriser le langage de programmation C et initier notre carrière par un aussi beau sujet.

également à nos professeurs des modules« Algorithmique » : Monsieur Ahmed ETTALBI pour le savoir qu'il nous a transmis et de « Techniques de programmation » : Monsieur Hatim GUERMAH pour les outils et les méthodes de programmation dispensées. Enfin, nos remerciements s'adressent aux professeurs et au corps administratif de l'Ecole Nationale Supérieure denl'Informatique et de l'Analyse des Systèmes-ENSIAS.

introduction

Dernièrement, beaucoup de gens reviennent aux outils informatiques afin de faciliter leurs vies quotidiennes. Ces outils peuvent être utilisés dans tous les secteurs, à savoir : le secteur professionnel, secteur éducatif et d'autres secteurs. La mise en oeuvre des techniques de l'informatique a été entreprise dans les bibliothèques, parce qu'elle apporte à terme la solution des principaux problèmes de gestion, la saisie des informations pour un prêt, paraissant simple, devait répondre à plusieurs options. Ces dernières étant obligatoires pour que l'emprunt d'un livre se fasse dans de bonnes conditions.Pour résoudre ce problème, on est obligé de finaliser un programme de gestion , qui sert à automatiser la Gestion des adhérents, des livres et des emprunts ce présent rapport retrace, à travers les différentes étapes que nous avons suivies pour atteindre cet objectif :

1.Le premier chapitre est consacré à l'analyse de la problématique et la conception du projet.

2.Le dernier chapitre aborde la réalisation du projet.

Table des matières

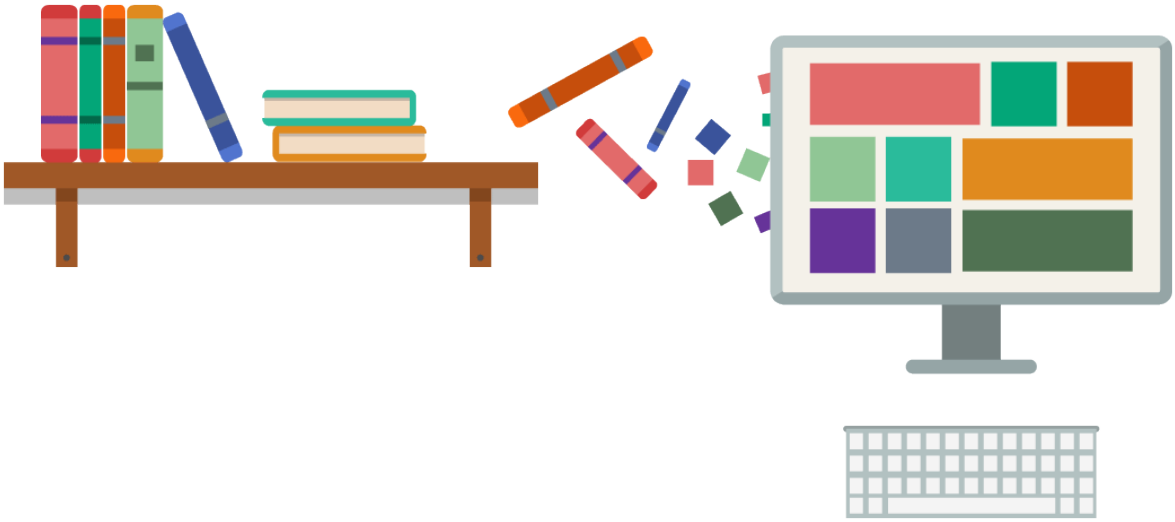
Chapitre 1

Présentation du projet

PROJET C s'agit de produire un programme d'environ 500 lignes de code afin de valider les compétences des cours : « Algorithmique », « Technique de programmation » et « Structures de données ». Le programme correspond à 20 heures de travail effectives en langage C. Les étudiants travaillent en binôme et bénéficient des conseils d'un professeur encadrant.

1.1 Sujet

notre projet s'agit de réaliserun système de gestion de bibliothèque, est un programme destiné à la gestion informatique des différentes activités nécessaires au fonctionnement d'une bibliothèque. Il permet notamment de gérer le prêt, la description, la consultation, la recherche et l'acquisition des livres.



Chapitre 2

Contexte général du projet

Ce chapitre permet de faire une analyse théorique du programme prêts de livre. En effet, cette conception est cruciale afin de comprendre la totalité des pricipes et les coder en se basant sur le cahier de charges fournit.

2.1 Cahier de charges :

Le cahier de charges présente l'ensemble des instructions et contraintes qui cadrent la réalisation du programme. Le cahier de charges disponible donne 3 instructions qu'on va élaborer dans cette partie.

2.2 problématique et objectifs :

développer une application en langage C pour la gestion de prêts de livres à la bibliotheque ENSIAS. Pour le faire, on va utiliser des structures de différentes natures, dont les champs contiennent les informations des livres et des emprunteurs.

2.2.1 Les besoins fonctionnels :

Le programme devra proposer à l'utilisateur un menu pour pouvoir naviguer entre les différentes fonctionnalités réalisées. En particulier la saisie des informations relatives aux adhérents et aux livres.

1.Gestion des adhérents :
Ajouter, Modifier, Supprimer un adhérent
Recherche d'un adhérent par son nom et affichage du résultat
Retour Menu Principal

2.Gestion des Livres :
Ajouter, Modifier, Supprimer un livre
Ordonner les livres par Catégorie
Recherche d'un livre par Catégorie et titre du livre.
Retour au Menu Principal

3.Gestion des emprunts :
Emprunter un livre
Afficher liste des livres empruntés
Afficher la liste des adhérents emprunteurs de livre
Rendre un livre

4.Quitter l'application

2.2.2 Les besoins non fonctionnels :

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et en ce qui concerne notre application, nous

avons dégagé les besoins suivants :

• **La disponibilité :**

L'application doit être disponible pour être utilisé par n'importe quel utilisateur.

• **La fiabilité :**

Les données fournies par l'application doivent être fiables. La possibilité de retourner au menu principal de l'application à partir de n'importe quelle fenêtre de celle-ci.

• **La performance :**

Le système doit réagir dans un délai précis, quel que soit l'action de l'utilisateur.

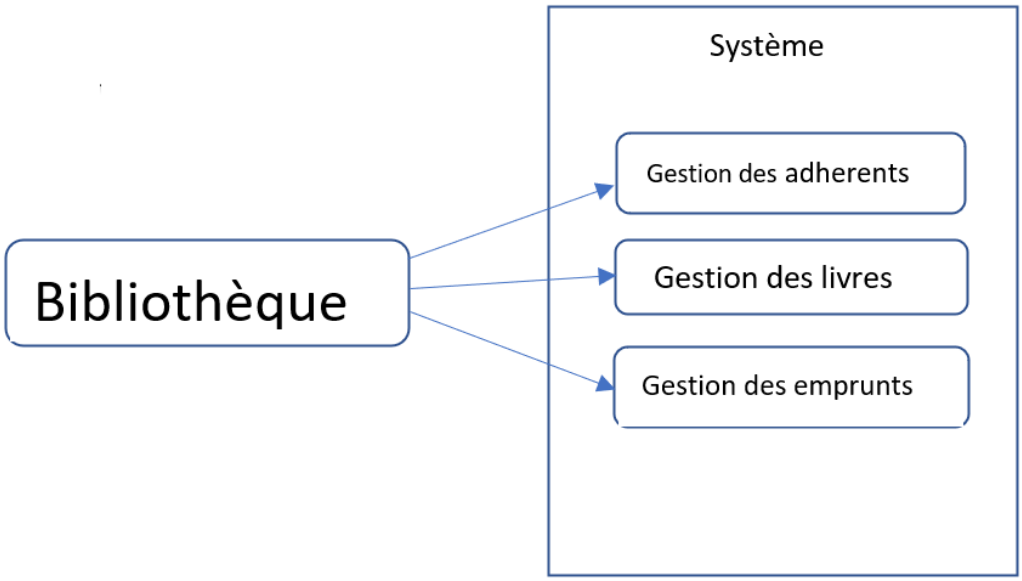
2.2.3 conclusion :

Après avoir dégagé les besoins fonctionnels et opérationnels et tous les critères qu'on doit prendre en considération, et afin de modéliser les besoin attendus de notre application et que les objectifs soient atteinte, on va suivre la démarche du processus unifié qu'on va le détailler dans la prochaine partie.

Chapitre 3

Analyse et conception

3.1 vue globale du système :



3.1.1 gestion des adherents

Description du traitement :
la getion des adhérents consiste à Ajouter, Modifier, Sup-

primer et Rechercher un adhérent. on illustre l'algorithme étulisé par l'organigramme suivant :

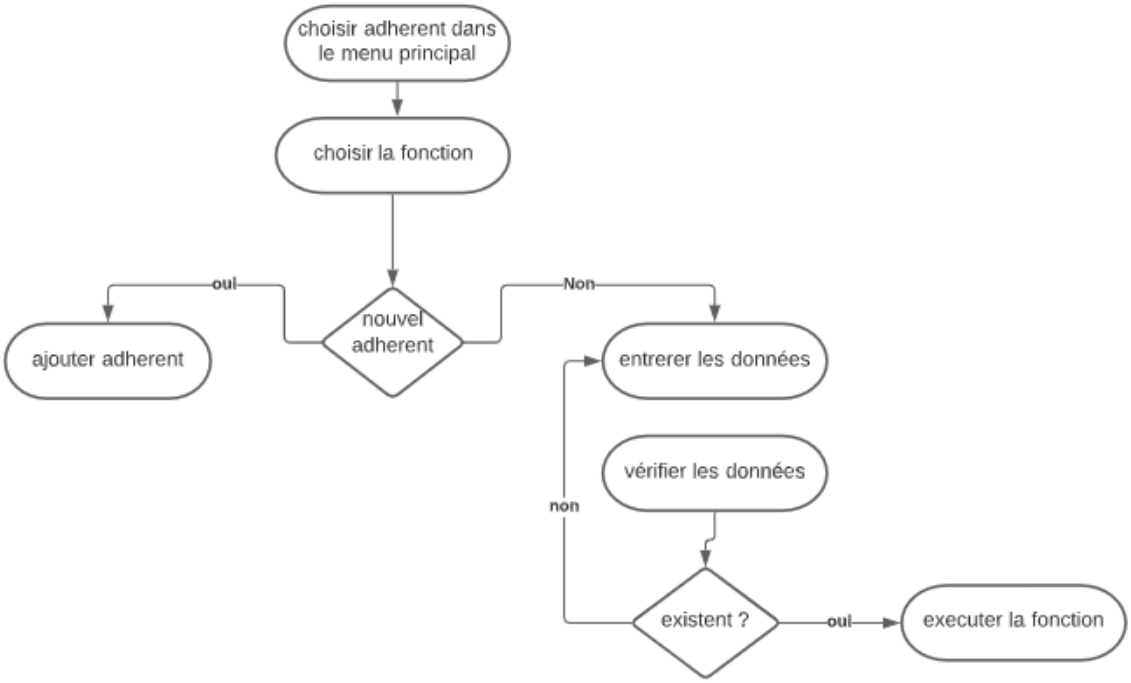


FIGURE 3.1 – Logo du logiciel LaTex

avec la possibilité de retour au menu .

3.1.2 gestion des livres

la gestion des livres consiste à : Ajouter, Modifier, Supprimer ,Ordonner et Recherche un livre. on illustre l'algorithme étulisé par l'organigramme suivant :

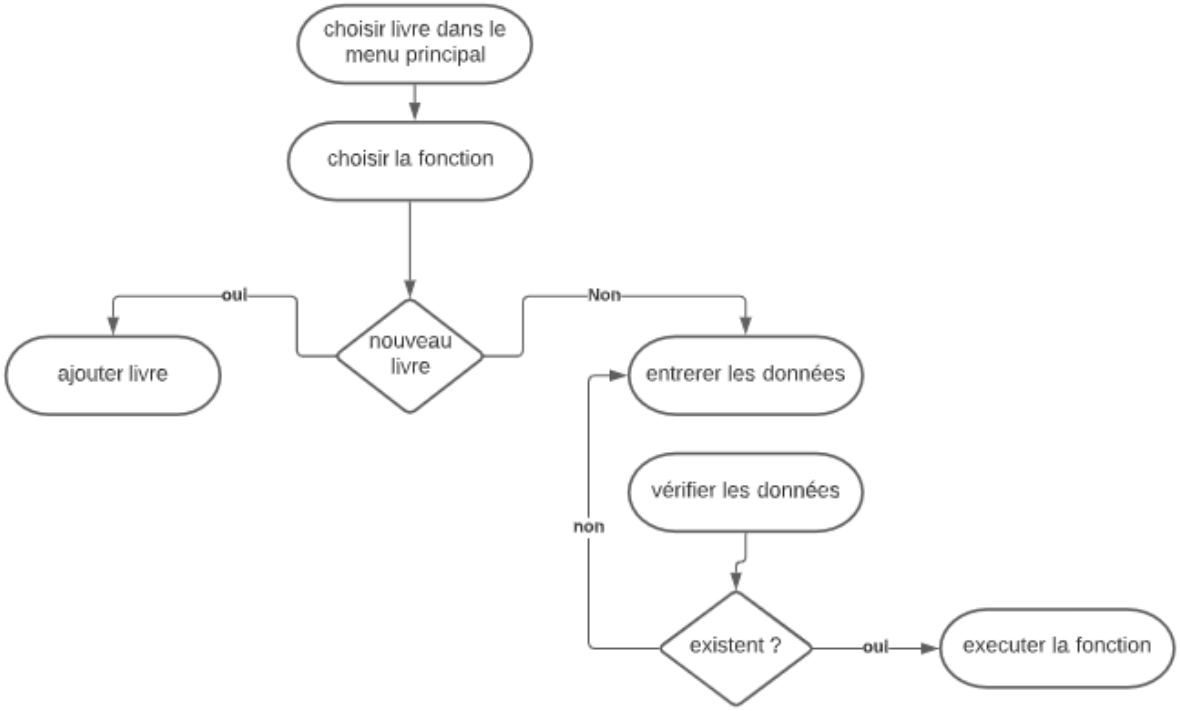


FIGURE 3.2 – organigramme gestion livres

3.1.3 gestion des emprunts

Description du traitement :
la getion desemprunts consiste à afficher les emprunteurs et les livres empruntés et réaliser l'emprunt et le rend des livres. les opérations : emprunter et rendre un livre necessitent la mise à jour des données du livre et d'emprunteur.

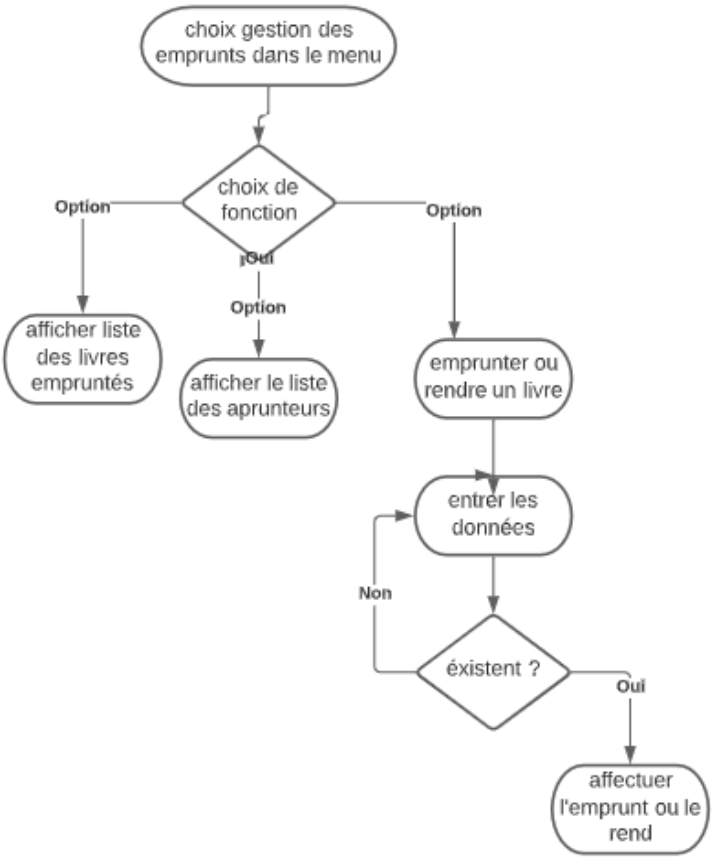


FIGURE 3.3 – organigramme gestion des emprunts

3.2 choix du langage

3.2.1 Rédaction du rapport :

La documentation professionelle nécessite la manipulation du logiciel de traitement de texte LaTex. Travailleravec ce dernier est inévitable tôt ou tard, donc nous avons voulu exploiter cette opportunité et explorer LaTex.

FIGURE 3.4 – Logo du logiciel LaTex

3.2.2 outils et thecniques de développement

Code : :Blocks est un environnement de développement intégré libre et multiplate-forme. Il est écrit en C++ et utilise la bibliothèque wxWidgets. Code : :Blocks est orienté C et C++, mais il supporte d'autres langages .

FIGURE 3.5 – Logo codeblocks

Chapitre 4

Réalisation et résultats

4.1 Difficultés rencontrées

Cette partie détaille la démarche de réalisation du projet et les problèmes qui l'accompagnaient.

4.1.1 manipulation des données

Parmi les choses les plus défiants dans ce projet précisément sont la manipulation des données des livres et des adhérents dans les fichiers texte, la manipulation de ces données serait plus facile ont utilisant SQL alors qu'il est imposé d'utiliser les fichiers pour stocker les données

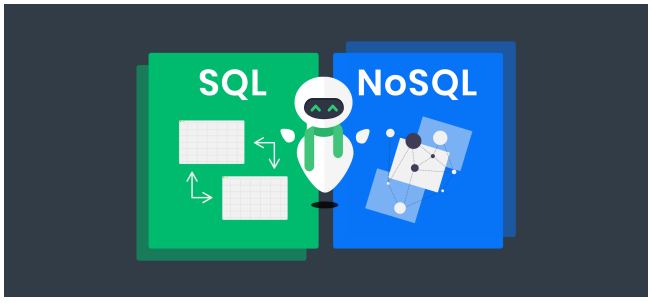


Figure 4.1 – SQL VS NO SQL

4.1.2 Programmation des fonctions

Programmer cette mini application s'accompagnait par plusieurs difficultés, car c'est notre première contact réel avec le langage C. Les erreurs de compilations étaient indispensables partout dans le code source.

4.1.3 Rédaction et organisation du code

Ecrire un clean code est un défi pour tous les développeurs. Nous avons essayé de garantir le maximum de la clareté. En effet, nous avons commenté le code d'une façon pertinente. De plus, nous avons divisé le code en plusieurs fichiers " .c " et des fichiers " .h ".

4.1.4 Contrainte de temps

La découverte de plusieurs technologies durant ce projet a consommé pas mal de temps. Nous pensons que le délai était suffisant mais trop serré.

4.2 Fonctions du code source

Comme vu précédemment dans le premier chapitre, chaque instruction necéssite une ou plusieurs fonctions. Dans cette partie nous allons élaborer chacune des fonctions utilisés dans le code source.

4.2.1 Les directives des préprocesseurs

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include "myfiles.h"
```

Ces directives ont comme role l'importation les bibliothèques et les fichiers header contenant les fonctions à utiliser dans le main.

4.3 fonctions gestion adhérents :

4.3.1 Fonction *ajouter* :

```
void ajout_adh()
```

cette fonction permet d'ajouter un nouvel adhérent
Input : les donneés du nouvel adhérent
Output :

4.3.2 Fonction *modifier* :

```
void modifierAdherent()
```

Input : les nouvelles donneés Output :

cette fonction permet de modifier un adhérent existant ,pour effectuer la modification dans notre fichier on a utilisé un fichier temporaire ,avant la modification la fonction verifie l'existence à l'aide de la fonction axiliaire :

```
bool recherche_adh(int numrech)
```

cette fonction auxiliaire permet de verifier l'existence d'un adherent à partir de son numéro. on a utiliser cette fonction auxiliaire et le fichier temporaire dans les fonctions siuvantes si necessaire.
Input : numéro d'adhérent
Output : true or false

4.3.3 Fonction *supprimer* :

```
void supp_adh()
```

cette fonction permet de supprimer un adhérent.
Input : le numéro d'adherent à supprimer
Output : void

4.3.4 Fonction *rechercher* :

```
void rech_adherent()
```

cette fonction permet de rechercher un adhérent par son nom,on cas d'existence de plusieurs adherent ayant le meme nom elle propose une recherche plus présise par prénom.
Input : le nom d'adherent recherché
Output : afficher les données d'adhérent ou introuvable s'il n'existe pas .

4.3.5 Fonction *rechercher* :

```
void rech_adherent()
```

cette fonction permet de rechercher un adhérent par son nom,on cas d'existence de plusieurs adherent ayant le meme nom elle propose une recherche plus présise par prénom.
Input : le nom d'adherent recherché
Output : afficher les données d'adhérent ou introuvable s'il n'existe pas .

4.3.6 Fonction *menu – adhrents* :

```
void affichageMenu_liv()
```

cette fonction permet de dériger l'utilisateur selon son choix de l'une des fonctions du gestion ahérents precedentes.
Input :
Output :void

4.4 fonctions gestion livres :

on traite la partie gestionlivre par analogie avec gestion adherent.

4.4.1 Fonction *ajouter* :

```
void ajout_liv()
```

cette fonction permet d'ajouter un nouveau livre
Input : les donneés du nouveau livre
Output : void

4.4.2 Fonction *modifier* :

```
void modifierlivre()
```

cette fonction permet de modifier un livre existant ,aprés verifier son existence à l'aide de la fonction axiliaire :

```
bool recherche_liv(int numrech)
```

cette fonction auxiliaire permet de verifier l'existence d'un livre à partir de son numéro.ON va l'utiliser dans les fonctions siuvantes si necessaire.

Input : numéro du livre
Output : true or false

4.4.3 Fonction *supprimer* :

```
void supp_liv()
```

cette fonction permet de supprimer un livre.
Input : le numéro du livre à supprimer
Output : void

4.4.4 Fonction *ordonner* :

```
void ordonner_Liv()
```

cette fonction permet d'ordonner les livres par catégorie,et par ordre alphabétique .
Input :
Output : afficher la liste des livres ordonnés

4.4.5 Fonction *chercher* :

```
void ordonner_Liv()
```

cette fonction permet de rechercher un livre par le tirtre ou par la catégoie selon le choix d'utilisateur.
Input : titre ou catégorie
Output : afficher les données du livre rechrché s'il existe ou introuvable s'il n'existe pas

4.4.6 Fonction $menu - livres$:

```
void affichageMenu_liv()
```

cette fonction permet de dériger l'utilisateur selon son choix de l'une des fonctions du gestion livres precedentes.
Input : choix d'utilisateur
Output :void

4.5 fonctions gestion emprunts :

4.5.1 Fonction $Emprunter - un - livre$:

```
void emprunter_livre()
```

cette fonction permet d'effectuer l'emprunt d'un livre , elle verifie que le livre est disponible (non emprunté), et que l'dherent ne dépasse pas le nombre maximal d'emprunts qui est trois livres .
Input : données du livre et d'emprunteurs
Output :void

4.5.2 Fonction $Affficher - liste - des - adhrents - emprunteurs$:

```
void afficher_liste_emp()
```

cette fonction permat d'afficher la liste des emprunteurs c'est à dire les adhérents qui ont emprunté au moins un livre .
Input :
Output :afficher la liste des emprunteurs

4.5.3 Fonction $Rendre - un - livre$:

```
void Rendre_livre()
```

cette fonction permet d'effectuer le rend d'un livre .
Input : données du livre et d'emprunteurs
Output :void

4.5.4 Fonction $Afficher - liste - des - livres - emprunts$:

```
void liste_livres()
```

cette fonction permet d'afficher la listes des livres empruntés .
Input :
Output :afficher la liste des livres empruntés

4.6 Fonction $affiche r$

afficher les données des livres et des adherents

```
void affichage_liv()
```

Input :
Output :afficher la liste des livres

```
void affichage_adh()
```

Input :
Output :afficher la liste des adherents

4.7 Fonction $main$

```

#include "myfiles.h"
int main()
{
```

```
    printf("\n\n\t\t\t---------------------------------- Le travail manuel est une
    lecture sans fin (pierre gascar)----------------------------------.\n\n");

     bienvenue();
    SetColor(35);
    menu_principal();
    merci();
return 0;
}
```

Cette fonction est le main de notre programme.
Input : void
Output :0

4.8 myfiles.h

```
// fonctions auxiliaires
bool recherche_adh(int numrech);
bool recherche_liv(int numrech);

// ajout
void ajout_adh();
void ajout_liv();
//suppression
void supp_adh();
void supp_liv();
//recherche
void rech_livre();
void rech_adherent();
//modification
void modifierAdherent();
void modifierlivre();

void ordonner_Liv();
//gestion des emprunts
void liste_livres();
void afficher_liste_emp();
void Rendre_livre();
void emprunter_livre();

//menu
void menu_principal();
void affichageMenu_adh();
void affichageMenu_liv();

//affichage
void affichage_adh();
void affichage_liv();


//style

void menu_principal();
void SetColor(int ForgC);
void merci();
void livre_sty();
```

Ce fichier .h contient une appelation de toutes les fonctions utilisées.

4.8.1 stockage des données

On a utilisé ici les fichiers textes pour stocker les données des adhérents et des livres. Un fichier texte est un fichier qui contient des informations sous la forme de caractères (en général en utilisant le code ASCII). Avantages d un fichier texte facile à manipuler facile à modifier (il suffit de prendre un éditeur de texte) facilement compréhensible par l utilisateur (les données sont représentées sous leur forme textuelle) portable (transférable d un ordinateur à

un autre sans grande difficulté)

Chapitre 5

mode d'emploi

5.1 Menu des choix :

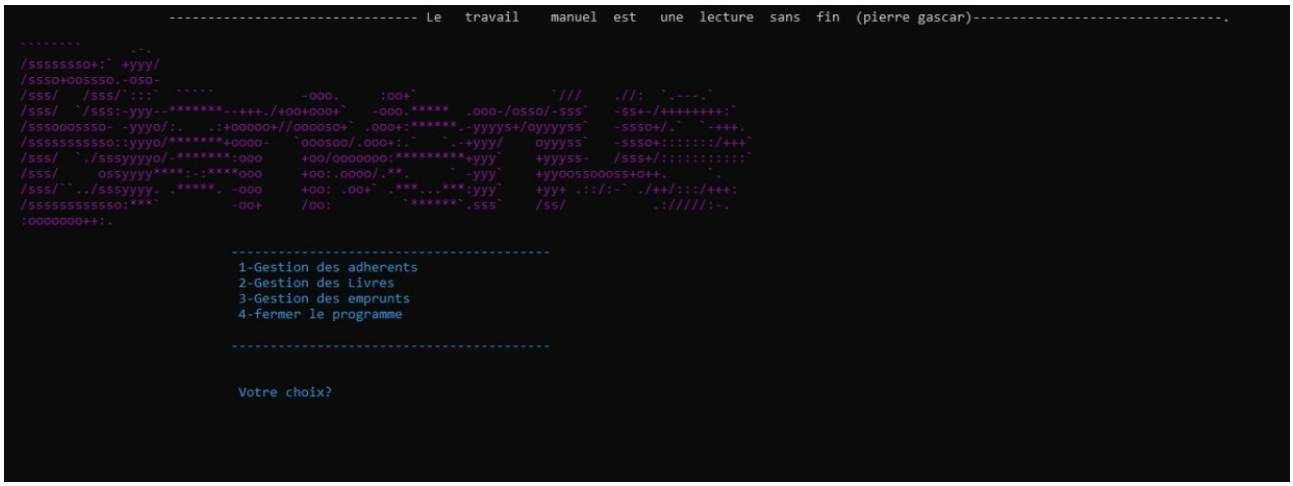


Figure 5.1 – MENU DES CHOIX

le menu donne quatre choix :
pour la gestion des adhérents entrer 1.
pour la gestion des livres entrer 2.
pour la gestion des emprunts entrer 3.
pour fermer le programme entrer 4.
selon le choix d'utilisateur l'utilisateur est dirigé vers l'un de ces menus :

5.2 Menu *livre* :

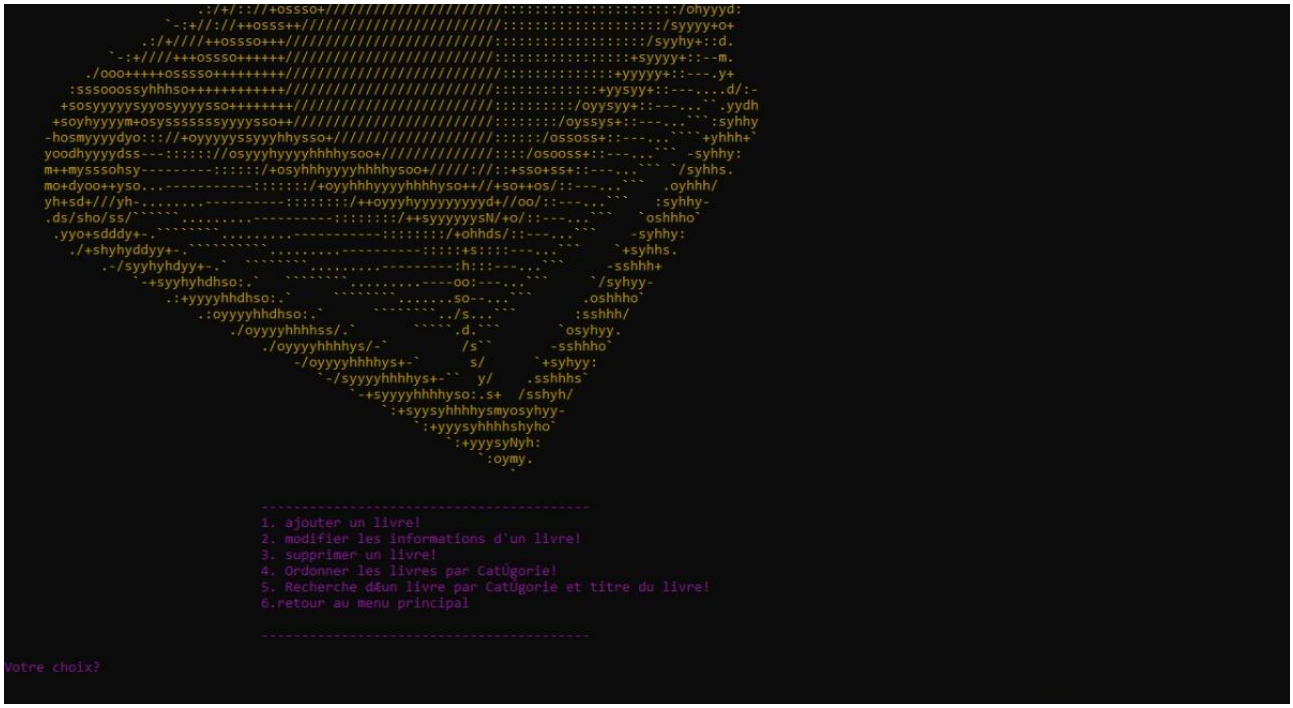


Figure 5.2 – MENU LIVRE

le menu livre donne six choix de fonctions :
pour ajouter un livre entrer 1.
pour mdifier les données d'un livre entrer 2.
pour supprimer un livre entrer 3.
pour ordonner les livres entrer 4.
pour rechercher un livre entrer 5.
pour revenir au menu principal entrer 6.

5.3 Menu *adhrent* :

Figure 5.3 – MENU ADHERENT

le menu adherent donne six choix de fonctions :
pour ajouter un adhérent entrer 1.
pour mdifier un adhérent d'un livre entrer 2.
pour supprimer un adhérent entrer 3.
pour rechercher un adhérent entrer 4.
pour revenir au menu principal entrer 5.

5.4 Menu *emprunts* :

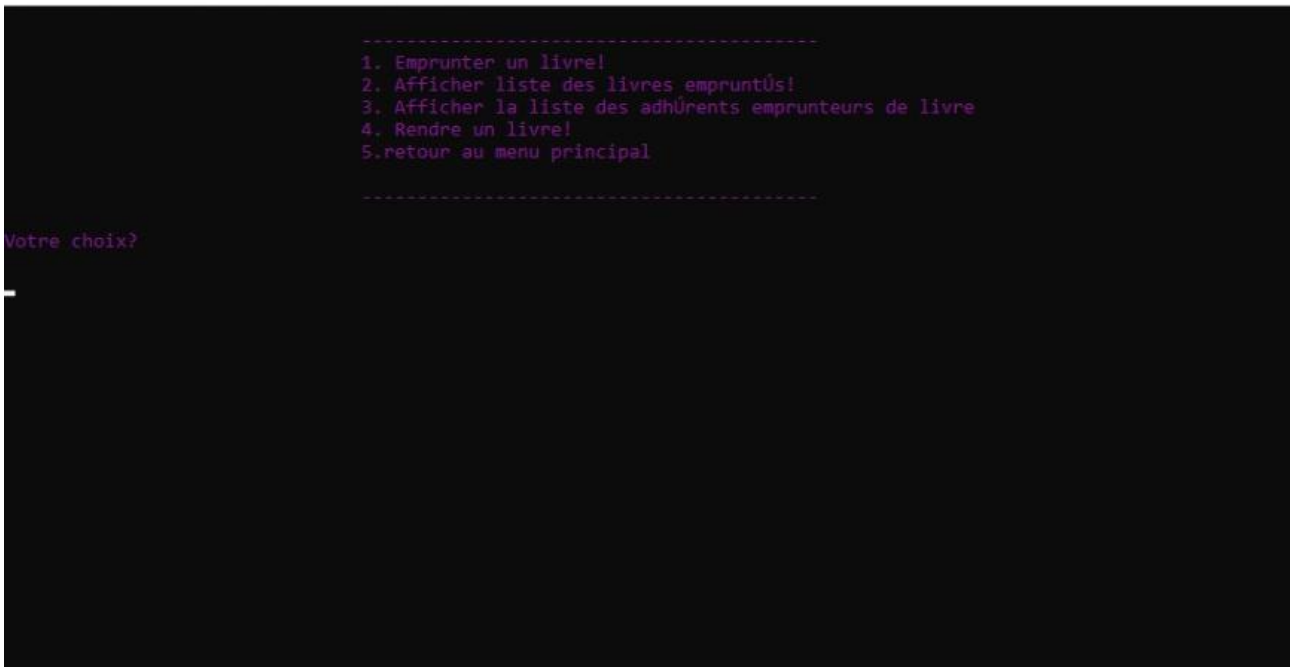


Figure 5.4 – MENU EMPRUNT

le menu emprunts donne six choix de fonctions :
pour emprunter un livre entrer 1.
pour afficher la liste des livres empruntés entrer 2.
pour afficher la liste des emprunteurs entrer 3.
pour reendre un livre entrer 4.
pour revenir au menu principal entrer 5.

Bibliographie

[1] <https://www.overleaf.com/project>, 2021. [Online ; accessed 18-January-2021].

[2] lucid. <http://lucid.app/>, 2021. [Online ; accessed 02-february-2021].

[3] Asir Mosaddek Sakib. C/c++ console application in code : :blocks. <http://www.http://blockofcodes.blogspot.com//>, 2013. [Online ; accessed 27-January-2021].

[4] Ludovic Vimont. Latex : Born to code : la programmation par l'exemple. <http://borntocode.fr/>, 2015. [Online ; accessed 16-January-2021].