|  |
| --- |
| [Nom de la société] |
| **Rapport du Mini-projet en langage C** |
| **Gestion des étudiants** |

|  |
| --- |
| **Réaliser par :SALIH safia et MELIANE ihssane**  **11/03/2035** |

[**Introduction** 2](#_Toc192590435)

[**Code Source** 3](#_Toc192590436)

[**Document de Conception** 14](#_Toc192590437)

[**Instructions d’installation et d’Exécution** 15](#_Toc192590438)

[**Tests et validation** 15](#_Toc192590439)

[**Conclusion** 17](#_Toc192590440)

[**Évaluation en Binôme** 17](#_Toc192590441)

# **Introduction**

Madame Hajar, nous souhaitons moi et ihssane MELIANE de vous présenter notre mini projet de programmation en C : un système de gestion des étudiants. Ce projet nous a permis d’appliquer nos connaissances en algorithmique et en manipulation des tableaux tout en renforçant notre capacité à structurer un programme.

La gestion des étudiants est essentielle dans le cadre d’un système d’information universitaire.

L’objectif principal est de permettre l’ajout, la gestion et l’affichage des informations des étudiants via un menu interactif. Ce projet représente une opportunité d’apprentissage enrichissante et nous a permis de surmonter divers défis techniques.

Et d’autres objectifs par exemple

• Consolider nos connaissances du langage C (variables, boucles, conditions, tableaux, etc.).

• Développer une application permettant la gestion des étudiants avec des fonctionnalités complètes.

• Améliorer notre logique de programmation et la gestion des structures de données.

• Apprendre à gérer des entrées utilisateur et à traiter des données efficacement.

## **Code Source**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[])

{

int NbEtud,i=0,j=0,c=0,ch,k,tr=0;

float moy[50],Notes[50][5],s=0,tnote[5],moyG;

char NomEtud[50][20],tnom[50],temp[50],Mtion[50][12],tm[15],rech[5];

printf("=================MENU=====================\n");

printf("1.Ajouter un etudiant\n2.Afficher la liste des etudiants\n3.Afficher le meilleur etudiants\n4.Rechercher un etudiants par nom\n5.Modifier un etudiant\n6.Trier les etudiants par moyenne\n7.Supprimer un etudiant\n8.Afficher les statistiques\n9.Filtrer les etudiants\n0.Quitter");

printf("\nVotre choix: ");

scanf("%d",&ch);

do{

switch(ch){

case 1 :

do{

printf("Entrez le nombre des etudiants ce que tu veux (limite 50 etudiants) : ");

scanf("%d",&NbEtud);

}while(NbEtud>50);

for(i=0;i<NbEtud;i++){

s=0;

c++;

printf("Entrez le nom de l'etudiant %d : ",i+1);

scanf(" %s",NomEtud[i]) ;

for(j=0;j<5;j++){

do{

printf("Entrez de la note %d de l'etudiant : ",j+1);

scanf("%f",&Notes[i][j]);

}while(Notes[i][j]<0 || Notes[i][j]>20);

s=s+Notes[i][j];

}

moy[i]=s/(float)5;

}

printf("%d etudiants sont ajoute avec succes ",c);

break;

case 2 :

for(i=0;i+1<NbEtud;i++){

// le tri alpha

if(strcmp(NomEtud[i],NomEtud[i+1])>0){

tr=1;

strcpy(temp,NomEtud[i]);

strcpy(NomEtud[i],NomEtud[i+1]);

strcpy(NomEtud[i+1],temp);

// la permutation des notes

for(j=0;j<5;j++){

tnote[i]=Notes[i][j];

Notes[i][j]=Notes[i+1][j];

Notes[i+1][j]=tnote[i];

//permutation de moyenne

temp[i]=moy[i];

moy[i]=moy[i+1];

moy[i+1]=temp[i];

}

}

for(j=0;j<NbEtud;j++){

if(moy[i]<10){

strcpy(Mtion[j],"Exclu");

}

else if(moy[i]<12){

strcpy(Mtion[j],"Passable");

}

else if(moy[i]<14){

strcpy(Mtion[j],"Assez bien");

}

else if(moy[i]<16){

strcpy(Mtion[j],"bien");

}

else if(moy[i]>=16){

strcpy(Mtion[j],"Tres bien");

}

}

}

// Affichage de la matrice

if(tr==1){

printf("Liste des etudiants et leurs notes :\n");

for (i=0;i<NbEtud;i++) {

printf("l'Etudiant : %s ,ses notes: ",NomEtud[i]);

for (j=0;j<5;j++) {

printf("%.2f, ", Notes[i][j]);

}

printf("le Moyenne : %.2f ,",moy[i]);

if(moy[i]<10)

printf(" Mention : Exclu");

else if(moy[i]<=12)

printf(" Mention : Passable");

else if(moy[i]<=14)

printf(" Mention : Assez bien");

else if(moy[i]<=16)

printf(" Mention : bien");

else if(moy[i]>16)

printf(" Mention : Tres bien");

printf("\n");

}

}

else

printf("il n'ya aucun etudiant enregistre");

break;

case 3 :

j=0;

for(i=0;i<NbEtud;i++){

if(moy[i]>moy[j]){

tr=1;

j=i;

}

}

if(tr==1){

printf("l'etudiant ayant le plus haute moyenne est %s avec un moyenne %.2f ",NomEtud[j],moy[j]);

}

else

printf("il n'ya aucun etudiant enregistre");

break;

case 4 :

printf("entrez le nom de l'etudiant : ");

scanf(" %s",&rech);

for(i=0;i<NbEtud;i++){

if(strcmp(NomEtud[i],rech)== 0){

tr=1;

printf("l'etudiant que tu recherche est : %s ,ses notes sont ",NomEtud[i]);

for (j=0;j<5;j++) {

printf("%.2f ,", Notes[i][j]);

}

printf("moyenne %.2f ,avec la",moy[i]);

if(moy[i]<10)

printf(" Mention : Exclu");

else if(moy[i]<12)

printf(" Mention : Passable");

else if(moy[i]<14)

printf(" Mention : Assez bien");

else if(moy[i]<16)

printf(" Mention : bien");

else

printf(" Mention : Tres bien");

printf("\n");

}

}

if(tr==0)

printf("l'etudiant n'existe pas");

break;

case 5 :

printf("saisir le nom de l'etudiant ");

scanf(" %s",&rech);

for(i=0;i<NbEtud;i++){

if(strcmp(NomEtud[i],rech)== 0){

tr=1;

s=0;

for(j=0;j<5;j++){

do{

printf("changer la note %d : ",j+1);

scanf("%f",&Notes[i][j]);

}while(Notes[i][j]<0 || Notes[i][j]>20);

s=s+Notes[i][j];

}

moy[i]=s/(float)5;

}

}

if(tr==0)

printf("l'etudiant n'existe pas");

break;

case 6 :

for(i=0;i<NbEtud-1;i++){

for(j=1;j<NbEtud;j++){

if(moy[i]<moy[j]){

tr=1;

tnote[i]=moy[i];

moy[i]=moy[j];

moy[j]=tnote[i];

}

}

}

if(tr==1){

printf("le tri decroissante par moyenne est : ");

for(j=0;j<NbEtud;j++){

printf("%.2f, ",moy[j]);

}

}

else

printf("il n'ya aucun etudiant enregistre");

break;

case 7 :

printf("Saisir le nom de l'etudiant a supprimer : ");

scanf("%s", rech);

tr = 0;

for (i = 0; i < NbEtud; i++) {

if (strcmp(NomEtud[i], rech) == 0) {

tr = 1;

// Decalage des etudiants

for (j=i;j<NbEtud-1;j++) {

strcpy(NomEtud[j], NomEtud[j + 1]);

moy[j] = moy[j + 1];

for(k=0;k<5;k++) {

Notes[j][k]=Notes[j+1][k];

}

}

// Reduction du nombre d'etudiants

NbEtud--;

printf("Suppression reussie.\n");

break;

}

}

if(tr==1)

printf("succes supprission ");

if(tr==0)

printf("l'etudiant n'existe pas");

break;

case 8 :

s=0;

c=0;

printf("le nombre des etudiant enregistre est : %d \n",NbEtud);

for(i=0;i<NbEtud;i++){

s=s+moy[i];

tr=1;

if(moy[i]>=10)

c++;

}

if(tr==1){

moyG=s/(float)NbEtud;

printf("le moyenne generale de classe est : %.2f \n",moyG);

printf("le nombre des etudiants ayant une moyenne superieur ou egale a 10 est : %d ",c);

}

if(tr==0)

printf("il n'ya aucun etudiant enregistre");

break;

case 9 :

tr=0;

for(i=0;i<NbEtud;i++){

if(moy[i]>=10){

tr=1;

printf("%s , ",NomEtud[i]);

}

}

if(tr==1)

printf("sont les étudiants ayant une moyenne superieur ou egale a 10 \n");

for(i=0;i<NbEtud;i++){

if(moy[i]<10){

tr=0;

printf("%s , ",NomEtud[i]);

}

}

if(tr==0){

printf("sont les etudiants ayant une moyenne inferieur a 10 ");

}

break;

default:

printf("invalide choix, Ressayez-vous autre fois .\n");

}

printf("\n=================MENU=====================\n");

printf("Votre choix : ");

scanf("%d",&ch);

}while(ch!=0);

return 0;

}

### **Document de Conception**

L’application repose sur une structure simple basée sur des tableaux pour stocker les étudiants et leurs notes. Les principales fonctionnalités implémentées sont :

• Ajout d’étudiants et stockage de leurs notes

• Calcul automatique de la moyenne

• Tri des étudiants par nom et par moyenne

• Recherche d’un étudiant par son nom

• Modification et suppression d’un étudiant

• Affichage des statistiques

Diagramme de flux du programme :

**[Démarrage]**

**|**

**[Menu Principal]**

**/ | \**

**[Ajout] [Affichage] [Quitter]**

#### **Instructions d’installation et d’Exécution**

#### 

Prérequis : Un compilateur C comme GCC.

Compilation et exécution :

gcc -o gestion\_etudiants gestion\_etudiants.c

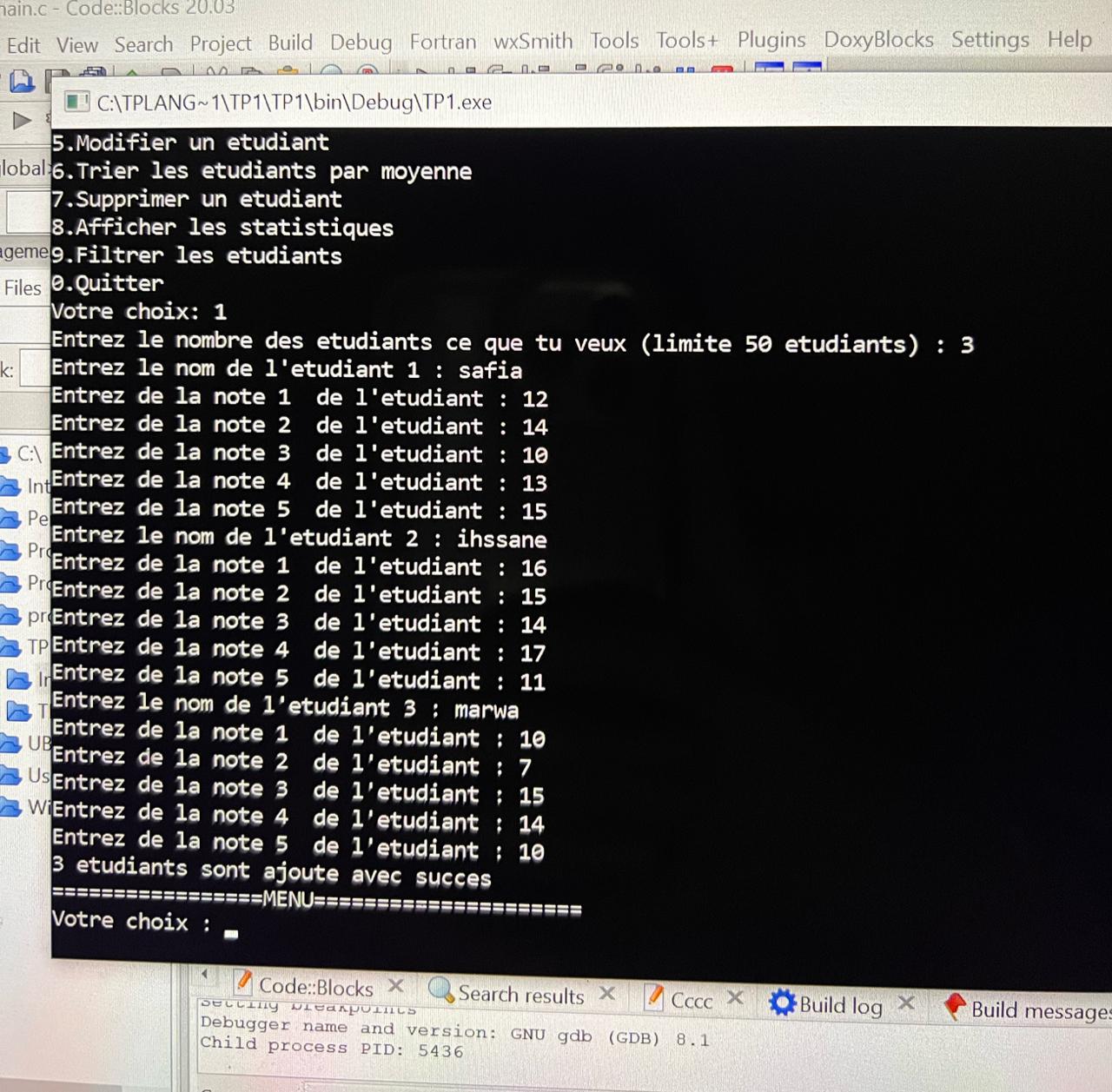
./gestion\_etudiants

##### **Tests et validation**

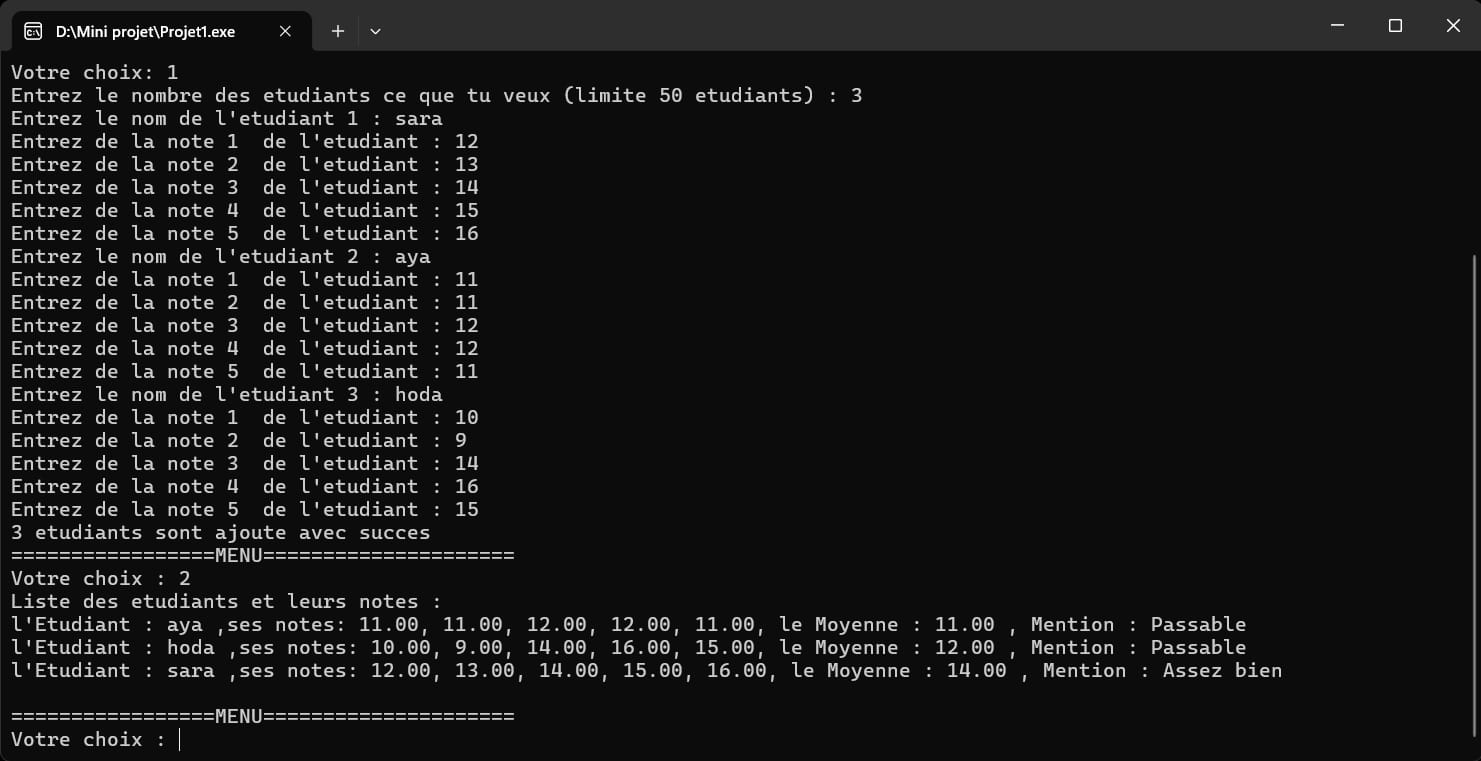
Le programme a été testé avec différents scénarios pour vérifier la robustesse et la gestion des erreurs.

Exemple de test :

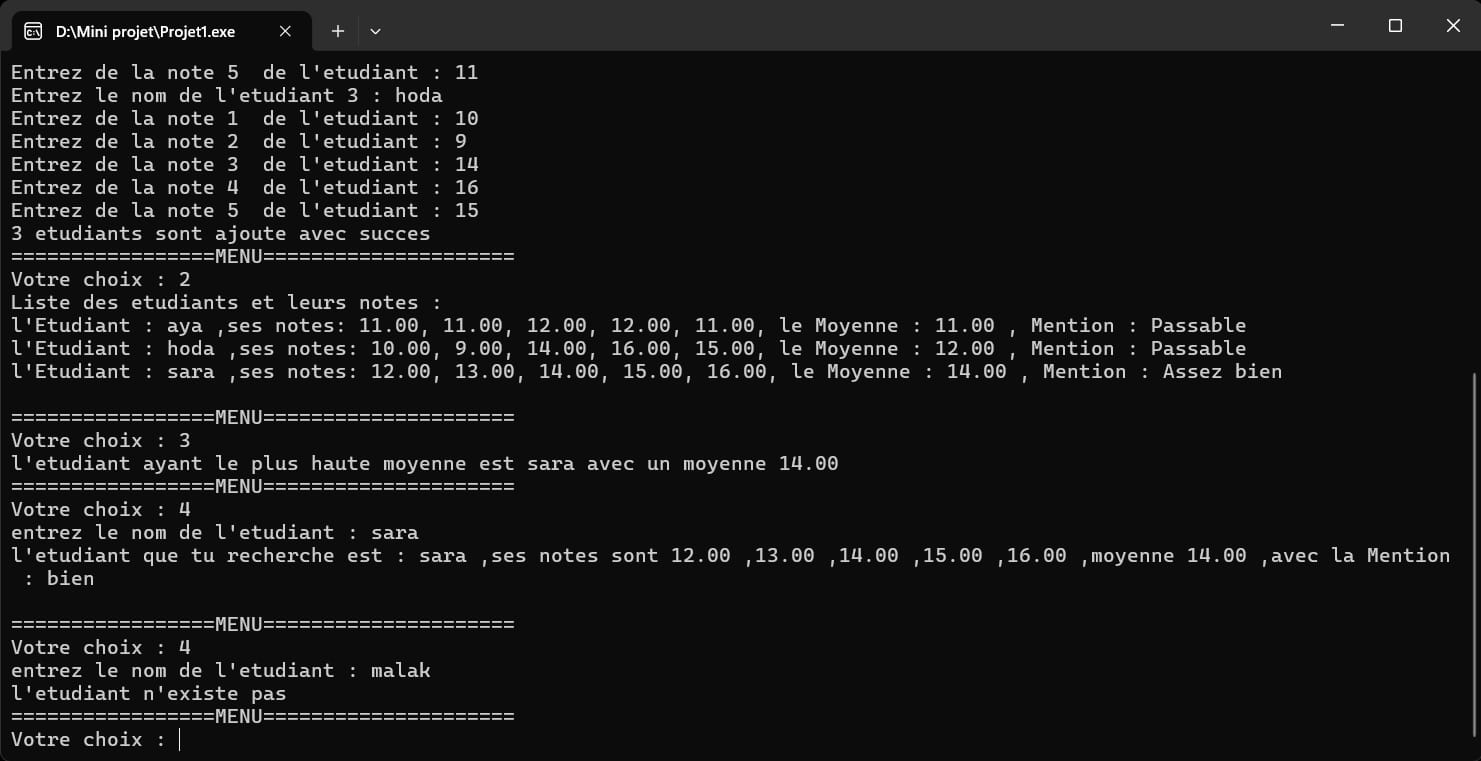
1. Ajout de 3 étudiants avec des notes valides.



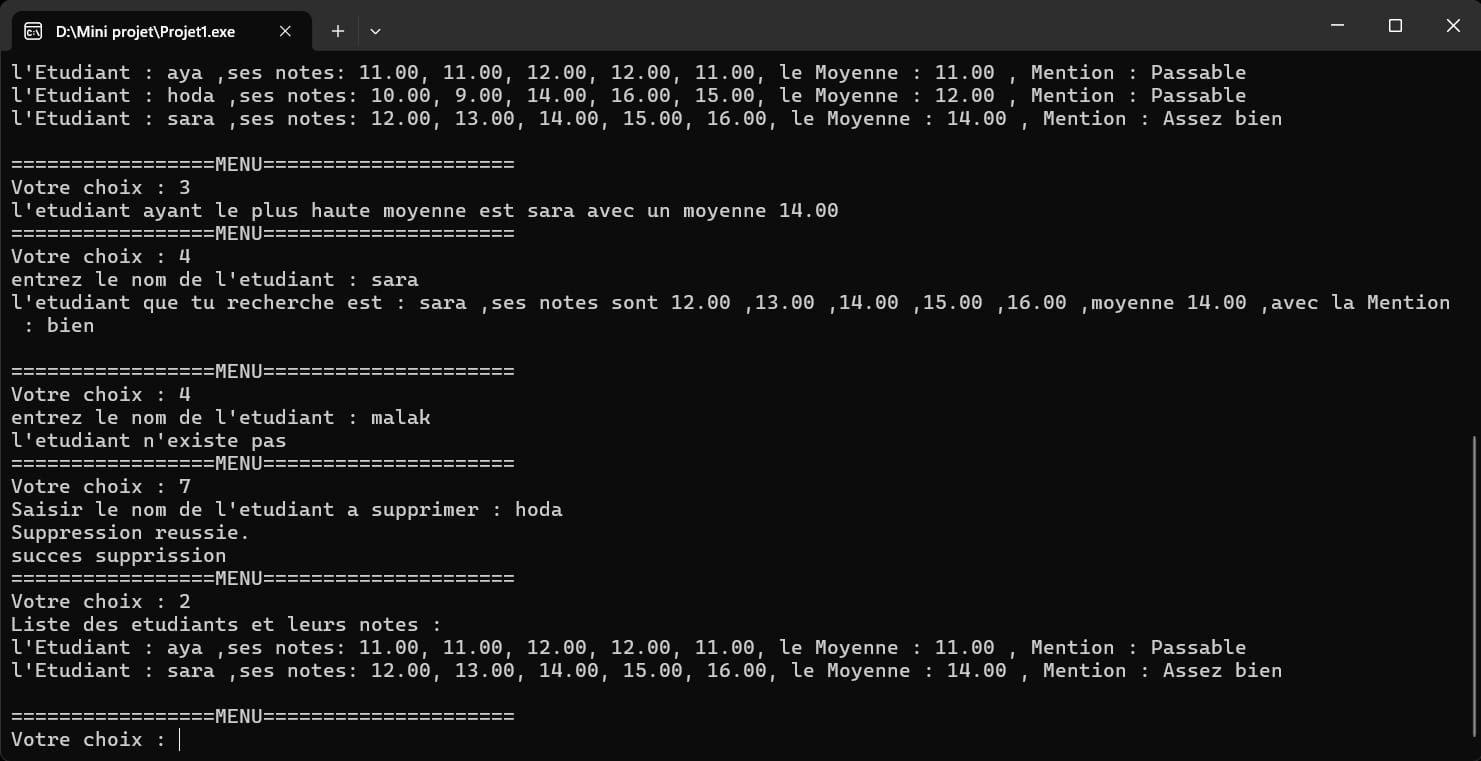
1. Affichage de la liste triée .



1. Recherche d’un étudiant existant et inexistant.



4.Suppression d’un étudiant et affichage de la nouvelle liste.



###### **Conclusion**

Ce projet a permis d’appliquer plusieurs concepts fondamentaux du langage C, notamment l’utilisation des tableaux, la gestion des chaînes de caractères et la programmation modulaire.

Des améliorations possibles incluent l’utilisation de structures pour mieux organiser les données et l’intégration de fichiers pour la persistance des informations.

**Évaluation en Binôme**

Le travail en binôme nous a permis de développer des compétences essentielles en programmation et en collaboration. Chacun de nous a apporté une contribution significative à la réalisation du projet, ce qui nous a permis d’optimiser notre organisation et notre efficacité.

Répartition des tâches :

• ihssane meliane : Conception de l’architecture du programme, implémentation des fonctionnalités de gestion des étudiants (ajout, suppression, modification) et optimisation du tri des données.

• safia salih : Développement de l’interface utilisateur, intégration des différentes fonctionnalités du menu interactif et validation du programme par des tests approfondis et la rédaction du rapport

Finalement permettez-moi de vous dire que Ce projet a été une expérience enrichissante qui nous a permis d’appliquer nos connaissances en C tout en développant notre capacité à structurer un programme efficace. Nous avons acquis de nouvelles compétences en gestion des tableaux, manipulation des chaînes et développement d’interfaces interactives.