|  |
| --- |
| Ecole Normale Supérieure de l'Enseignement Technique - Mohammedia |
| Compte Rendu TP Synthèse |
| Structures de données |
|  |
|  |
| **Enseignant : M.A.NAJI Par : Moustaid Ayoub** |

|  |
| --- |
| Année universitaire : 2014/2015 |

Code source : (partie1)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//Déclaration constantes

#define NBETUDIANT 2

#define NBMODULE 2

//Déclaration structures

typedef struct Module{

int num;

char \*intitule;

float coef;

}Module;

typedef struct Resultat\_Module{

int num\_module;

float \*moyenne;

char \*valide;

}Resultat\_Module;

typedef struct Etudiant{

char \*nom;

char \*prenom;

float resultat\_semestre;

Resultat\_Module \*result;

}Etudiant;

//prototypes

void initialiser\_etudiants(Etudiant[]);

void initialiser\_modules(Module[]);

void saisir\_notes(Etudiant[],Module[]);

void afficher\_resultat(Etudiant[]);

int calcul\_resultat\_semestre(Etudiant[],Module[]);

void afficher\_resultat\_semestre(Etudiant[]);

int verifier\_note(Etudiant[]);

void afficheResultatEtudiant(Etudiant);

int rechercher\_etudiant(Etudiant[],char\*);

void supprimerEtudiant(Etudiant[],int);

void exporter(Etudiant[],Module[]);

void initialiser\_etudiants(Etudiant tabEt[]){

fflush(stdin);

int i,j;

char nom[32];

char prenom[32];

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

printf("Nom :");

gets(nom);

printf("Prenom :");

gets(prenom);

tabEt[i].nom=(char\*)malloc(sizeof(nom));

tabEt[i].prenom=(char\*)malloc(sizeof(prenom));

tabEt[i].result=(Resultat\_Module\*)malloc(sizeof(Resultat\_Module)\*NBMODULE);

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

\*(tabEt[i].result[j].moyenne)=-1;

}

strcpy(tabEt[i].nom,nom);

strcpy(tabEt[i].prenom,prenom);

}

}

void initialiser\_modules(Module tabMod[]){

fflush(stdin);

int i;

char intitule[50];

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

printf("Le Numero de Module :");

scanf("%d",tabMod[i].num);

printf("l'intitule de Module' :");

scanf("%s",intitule);

printf("le Coef de Module' :");

scanf("%f",tabMod[i].coef);

tabMod[i].intitule=(char\*)malloc(sizeof(intitule));

strcpy(tabMod[i].intitule,intitule);

}

}

void saisir\_notes(Etudiant tabEt[],Module tabMod[]){

int i,j;

float moyenne;

char \*buf=malloc(sizeof(float));

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

printf("\*\*\*\*\*\*\*\*\*\* %s \*\*\*\*\*\*\*\*\*\*\n",tabMod[j].intitule);

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

if(\*(tabEt[i].result[j].moyenne)!=-1){

printf("%f (Entrer pour confirmer)\n",\*(tabEt[i].result[j].moyenne));

}

else{

fflush(stdin);

gets(buf);

if(strlen(buf)!=0)

moyenne=atof(buf);

else

moyenne=-1;

}

tabEt[i].result[j].num\_module=tabMod[j].num;

\*(tabEt[i].result[j].moyenne)=moyenne;

tabEt[i].result[j].valide=(char\*)malloc(sizeof((moyenne>=12?"v":"nv")));

strcpy(tabEt[i].result[j].valide,(moyenne>=12?"v":"nv"));

}

}

}

void afficher\_resultat(Etudiant tabEt[]){

int j;

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

afficheResultatEtudiant(tabEt[j]);

}

}

int calcul\_resultat\_semestre(Etudiant tabEt[],Module tabMod[]){

if(verifier\_note(tabEt)==1){

int i,j;

float resultat\_semestre=0;

float somme\_coef=0;

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

somme\_coef+=tabMod[i].coef;

}

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

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

resultat\_semestre+=\*(tabEt[j].result[i].moyenne)\*tabMod[i].coef;

}

resultat\_semestre/=somme\_coef;

tabEt[j].resultat\_semestre=resultat\_semestre;

}

return 1;

}

else

return -1;

}

void afficher\_resultat\_semestre(Etudiant tabEt[]){

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Resultat de Semestre \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

int i;

float moy\_classe=0,note\_max,note\_min;

int nb\_v=0,nb\_nv=0;

note\_max=tabEt[0].resultat\_semestre;

note\_min=tabEt[0].resultat\_semestre;

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

printf("%s \t %s \t",tabEt[i].nom,tabEt[i].prenom);

printf("%f\n",tabEt[i].resultat\_semestre);

moy\_classe+=tabEt[i].resultat\_semestre;

(tabEt[i].resultat\_semestre>=12)?nb\_v++:nb\_nv++;

if(tabEt[i].resultat\_semestre>note\_max)

note\_max=tabEt[i].resultat\_semestre;

if(tabEt[i].resultat\_semestre<note\_min)

note\_min=tabEt[i].resultat\_semestre;

}

moy\_classe/=NBETUDIANT;

printf("\n...\n\n");

printf("Moyenne de Classe : %f\n",moy\_classe);

printf("Note Max : %f\n",note\_max);

printf("Note Min : %f\n",note\_min);

printf("Nombre d'etudiants ayant validé le Semestre : %d\n",nb\_v);

printf("Nombre d'etudiants n'as pas validé le Semestre : %d\n",nb\_nv);

}

int verifier\_note(Etudiant tabEt[]){

int i,j;

for(i=0;i<NBETUDIANT;i++)

for(j=0;j<NBMODULE;j++)

{

if(\*(tabEt[i].result[j].moyenne)==-1)

{return -1;}

}

return 1;

}

void afficheResultatEtudiant(Etudiant etudiant){

printf("%s \t %s \t",etudiant.nom,etudiant.prenom);

int i;

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

if(\*(etudiant.result[i].moyenne)!=-1)

printf("%f\t",\*(etudiant.result[i].moyenne));

else

printf("\t");

}

printf("\n");

}

int rechercher\_etudiant(Etudiant tabEt[],char \*nom){

int i;

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

if(strcmp(nom,tabEt[i].nom)==0)

return i;

}

return -1;

}

void supprimerEtudiant(Etudiant tabEt[],int ind){

int i,j;

for(i=0;i<NBETUDIANT;i++)

{

if(strcmp(tabEt[i].nom,tabEt[ind].nom)==0)

{

int c;

printf("\n confirmez suppression : tapez 1 pour confirmez ou 2 pour annuler ? \n");

scanf("%d",&c);

if(c==1){

free(tabEt[i].result);

for(j=i;j<NBETUDIANT-1;j++)

tabEt[j]=tabEt[j+1];

puts("Suppression effectuee !");

break;

}

else

break;

}

}

}

void exporter(Etudiant tabEt[],Module tabMod[]){

FILE\* fichier = NULL;

int i,j;

fichier = fopen("fichier.txt", "w");

if (fichier != NULL)

{

fprintf(fichier,"Nom \t Prenom \t");

for(i=0;i<NBMODULE;i++)

fprintf(fichier,"%s\t",tabMod[i].intitule);

fprintf(fichier,"\n");

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

fprintf(fichier, "%s\t%s\t", tabEt[i].nom,tabEt[i].prenom);

for(j=0;j<NBMODULE;j++)

if(\*(tabEt[i].result[j].moyenne)!=-1)

fprintf(fichier, "%f\t", \*(tabEt[i].result[j].moyenne));

else

fprintf(fichier,"Vide \t");

fprintf(fichier,"\n");

}

fclose(fichier);

system("fichier.txt");

}

}

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

int et=0,mod=0,not=0;

Etudiant tabEt[50];

Module tabMod[20];

char nom[32];

int numero;

int choix;

do{

printf("\n\n------------------------------ Menu ------------------------------\n\n");

puts("1. Saisie des Etudiants");

puts("2. Saisie des Modules");

puts("3. Saisie des Notes Module'");

puts("4. Details d'un Etudiant");

puts("6. Afficher Les resultats de tous les Etudiants");

puts("7. Afficher Les resultats d'un Semestre");

puts("8. Supprimer un Etudiant");

puts("9. Exporter vers Fichier");

puts("0. Quitter");

printf("\n------------------------------ Menu ------------------------------\n");

printf("\n Votre choix : ");

scanf("%d",&choix);

switch(choix){

case 1 : et=1; initialiser\_etudiants(tabEt); break;

case 2 : mod=1; initialiser\_modules(tabMod); break;

case 3 : if(et==1&&mod==1)

{ not=1; saisir\_notes(tabEt,tabMod);}

else

puts("Tableau Etudiants/Modules vide !! ");

break;

case 4 :

if(et==1)

{

printf("Nom etudiant : ");

scanf("%s",nom);

int ind = rechercher\_etudiant(tabEt,nom);

if(ind!=-1)

afficheResultatEtudiant(tabEt[ind]);

else

puts("Etudiant Innexistant !!");

}

else

puts("Tableau des Etudiant Vide");

break;

case 6 : if(et==1&&mod==1)

afficher\_resultat(tabEt);

else

puts("Tableau Etudiants/Modules Vide !!");

break;

case 7 : if(et==1&&mod==1&&not==1){

if(calcul\_resultat\_semestre(tabEt,tabMod)==1)

afficher\_resultat\_semestre(tabEt);

else

puts("Des notes introuvables !! ");

}

else

puts("Tableau d'etudiants/Modules Vide");

break;

case 8 : if(et==1){

printf("Nom etudiant : ");

scanf("%s",nom);

int ind = rechercher\_etudiant(tabEt,nom);

if(ind=!-1)

supprimerEtudiant(tabEt,ind);

else

puts("Etudiant Innexistant !! ");

}

else

puts("Tableau d'etudiants vide !!");

break;

case 9 : if(et==1&&mod==1)

exporter(tabEt,tabMod);

else

puts("Tableau Etudiants/Modules Vide");

break;

case 0 : exit(-1); break;

default : puts("Erreur choix !");

}

}while(choix=!0);

return 0;

}

Partie2

# Questions

1-La mise en place en utilisant les tableaux ne présent pas des grands difficultés sauf parfois au niveau de la compilation.

2-Exemple de shéma mémoire :

Structure étudiant au niveau de la mémoire

Nom

Prenom

valide

Moyenne

|  |
| --- |
| @(pointeur sur nom) |
| @(pointeur sur prenom) |
| @(pointeur sur result) |

|  |
| --- |
| Num\_module |
| @(pointeur sur moyenne) |
| @(pointeur sur valide) |

Structure Result\_Module au niveau de la mémoire