

Dossier de développement logiciel

SAE S1.01 – Initiation au développement Implémentation d'un besoin client

Table des matières

1. Présentation du projet.	3
a. Le rôle fonctionnel de l'application.	3
b. Les entrées et sorties de l'application.	3
2. Organisation.	4
a. Organisation des tests de l'application.	4
b. Bilan de validation des sprints.	4
3. Bilan de projet.	5
a. Difficultés rencontrées.	5
b. Ce qui est réussi.	6
c. Ce qui peut être amélioré.	6
Annexe du dossier	7
- Listing complet des sources	7
- Trace d'exécution du test du sprint le plus haut	26
- Trace d'exécution d'un de nos jeux de test du sprint le plus haut	29

1. Présentation du projet.

a. Le rôle fonctionnel de l'application.

Ce projet est une application ayant pour but de gérer une formation universitaire. Elle permet d'ajouter et de visualiser les notes de chaque étudiant sous forme de relevé de note. Ainsi, elle a pour rôle de décider du passage ou non, d'un étudiant à l'année suivante, grâce à l'analyse des moyennes de chaque UE.

b. Les entrées et sorties de l'application.

- Pour sortir du programme – Commande 1:

IN : exit | OUT : arrêt de l'exécution.

- Pour définir le nombre d'UE – Commande 2 :

IN : formation + nombre d'UE | OUT : « Le nombre d'UE est défini » si le nombre d'UE est correct ($3 < UE < 6$), sinon « Le nombre d'UE est incorrect » ou sinon « Le nombre d'UE est déjà défini »

- Pour ajouter une épreuve – Commande 3:

IN : épreuve + numéro de semestre + nom de la matière + nom de l'épreuve + coefficients par UE | OUT : « Matière ajoutée à la formation » + « Epreuve ajoutée à la formation » sinon « Une même épreuve existe déjà » ou « Au moins un des coefficients est incorrect » ou encore « Le numéro de semestre est incorrect ».

- Pour vérifier les coefficients – Commande 4:

IN : coefficients + numéro de semestre | OUT : « Coefficients corrects » sinon « Les coefficients d'au moins une UE de ce semestre sont tous nuls » ou encore « Le semestre ne contient aucune épreuve ».

- Pour ajouter une note – Commande 5 :

IN : note + numéro de semestre + nom de l'étudiant + nom de la matière + nom de l'épreuve + note | OUT : « Etudiant ajoutée à la formation » + « Note ajoutée à l'étudiant » ou sinon l'un des messages d'erreurs : « Matière inconnue », « Epreuve inconnue », « Note incorrecte », « Une note est déjà définie pour cet étudiant ».

- Pour vérifier les notes – Commande 6:

IN : notes + numéro de semestre | OUT : « Notes correctes » ou sinon l'un des messages d'erreurs : « Le numéro de semestre est incorrect », « Etudiant inconnu », « Il manque au moins une note pour cet étudiant ».

- Pour afficher le relevé de notes d'un étudiant - Commande 7:

IN : releve + numéro de semestre + nom de l'étudiant | OUT : affichage du relevé si tout est correct ou l'un des messages d'erreurs « Le numéro de semestre est incorrect », « Etudiant inconnu », « Les coefficients de ce semestre sont incorrects », « Il manque au moins une note pour cet étudiant ».

- Pour afficher la décision – Commande 8 :

IN : decision + nom de l'étudiant | OUT : « Passage » si toutes les moyennes d'UE sont supérieures ou égales à 10 ou « Aucune » si elles sont inférieures à 10 ou « Redoublement » si aucune des 2 conditions sont atteintes ou « Etudiant inconnu » ou « Les coefficients d'au moins un semestre sont incorrects » ou « Il manque au moins une note pour cet étudiant ».

2. Organisation.

a. Organisation des tests de l'application.

Pour organiser les tests de notre application, nous sommes passés par différentes étapes :

- Réflexion sur le fonctionnement des commandes.
- Schématisation des étapes des commandes.
- Ecriture en pseudo-code.
- Création du code.
- Tests simples sur quelques commandes.
- Résolutions de bugs.
- Test de l'application avec les différents jeux de test du professeur.
- Test de l'application avec notre propre jeu de test.
- Correction des derniers bugs.

b. Bilan de validation des sprints

Au total, tous les sprints ainsi que le sprint bonus ont été validés le jeudi 10 novembre 2022 : le sprint 4 a été validé avec les alignements contrairement au sprint bonus qui a été validé sans alignements. Entre-temps, nous avons amélioré le code. Maintenant, tous les sprints y compris le sprint bonus fonctionnent correctement avec les alignements.

Ci-dessous, vous pouvez visualiser notre planning de validations des sprints :

- Semaine 1 – 3 : validation du sprint 1 (C1 à C4).
- Semaine 4 – 5 : validation du sprint 2 (C5 à C6).
- Semaine 6 – 7 : validation du sprint 3 (C7).
- Semaine 7 – 8 : validation du sprint 4 (C8).

3. Bilan de projet.

a. Difficultés rencontrées.

Tout au long de ce projet, nous avons rencontré plusieurs difficultés que nous avons réussi à surmonter. Ci-dessous, vous retrouverez dans l'ordre toutes les difficultés rencontrées :

- Problème de lecture de la commande tapé : nous sommes parti dans l'optique de lire toute une ligne au lieu d'un seul paramètre à la fois. Pour cela, nous avons essayé avec des « fgets » et « scanf("%[^\n]", cmd); ». Toutefois, nous nous sommes rendus compte que ce raisonnement avait ses limites. Ainsi, nous avons pensé différemment en nous rappelant que l'on pouvait lire plusieurs paramètres à l'aide d'un seul « scanf ».
- Difficulté à faire les structures : nous nous sommes inspirés des structures du professeur plus tard.
- Difficulté de lecture des épreuves dans des structures, en utilisant des pointeurs dans une fonction : nous mettions la flèche à des endroits où il ne fallait pas la mettre.
- Difficulté à trouver la bonne structure étudiant : nous ne savions pas où la placer, nous avons même l'idée de faire une structure Promotion afin de mettre la structure étudiant à l'intérieur.
- Problème pour afficher 1 seule fois les messages d'erreurs. Pour cela, nous avons créé une variable « nbEtuTemp » qui, après que la commande est affichée le message d'erreur, incrémente nbEtuTemp. Comme nbEtuTemp sera égal à 1, alors le message d'erreur ne sera pas affiché une autre fois.
- Difficulté à faire la somme des coefficients par UE.
- Difficulté à faire les arrondissements des notes.
- Difficulté pour les alignements du relevé.
- Problème du main qui prenait trop de place mémoire. Pour le résoudre, nous avons créé des fonctions afin d'alléger le code.

b. Ce qui est réussi.

Malgré quelques difficultés, la lecture des commandes fonctionne correctement, les structures sont optimisées et fonctionnelles, le calcul des moyennes renvoie un résultat correct et cohérent. Également, la documentation de quelques fonctions sont assez bien réussies. Ainsi, tous les sprints ont été réussis et validés avec le bonus.

c. Ce qui peut être amélioré.

Certes, tous les sprints ont été validés mais il reste tout de même certaines améliorations qui peuvent être effectuées. Par exemple, pour optimiser le code, nous pourrions créer des fonctions puisque nous avons encore des répétitions de lignes de code. Cela augmenterait également la rapidité de l'exécution, d'autant plus si le main contient moins de ligne que possible. La clarté du code peut aussi être améliorée afin d'être d'avantage compréhensible dès la première lecture.

Annexe du dossier

- Listing complet des sources

Le code :

```
/**
 * @file SAE_S1_01_Namodacane_Alexandre.c
 * Application de gestion de formation universitaire
 * @author KALIAMOORTHY Namodacane et CAROUNANITHI Alexandre
 * Date de création : 30/09/2022
 * @version 8.7 11/11/2022
 * Codé sur Visual Studio 2022
 */

#include <stdio.h>
#include <string.h>
#include <assert.h>
#include <math.h>
#pragma warning(disable : 4996 6031 6262 4244 6054 6385)

//Declaration de constante
enum {
    NB_SEMESTRES = 2,
    MIN_UE = 3,
    MAX_UE = 6,
    MAX_MATIERES = 10,
    MAX_EPREUVES = 5,
    MAX_ETUDIANTS = 100,
    MAX_CHAR = 30,
};
typedef enum { true = 1, false = 0 }bool;
const float MIN_NOTE = 0.f, MAX_NOTE = 20.f;
typedef char CH30[MAX_CHAR];

//Definition de structures
typedef struct {
    //Structure pour sauvegarder les coefficients de chaque UE en fonction de
    //l'épreuve et les notes de chaque UE en fonction de l'épreuve
    CH30 nom;
    float coef[MAX_UE];
    float note[MAX_UE];
}Epreuve;

typedef struct {
    //Structure pour sauvegarder les épreuves pour chaque Matiere
    CH30 nom;
    unsigned int nbEpreuves;
    unsigned int nbNotes;
    Epreuve epreuves[MAX_EPREUVES];
}Matiere;

typedef struct {
```

```

//Structure pour sauvegarder les matieres pour chaque Semestre
    unsigned int nbMatiere;
    Matiere matieres[MAX_MATIERES];
    float moyenneNotes[MAX_UE];
}Semestre;

typedef struct {
//Structure pour sauvegarder les noms des étudiants
    CH30 nom;
    Semestre semestres[NB_SEMESTRES];
}Etudiant;

typedef struct {
//Structure pour sauvegarder les semestres et leurs UE respectives
    unsigned int nbUE;
    unsigned int nbEtudiant;
    Etudiant etudiants[MAX_ETUDIANTS];
}Formation;

//Les fonctions
void init(Formation* f) {
/**
 * Initialise les valeurs des structures à 0 ou -1
 * @param[in] Structure Formation f
 * @param[out] Structure Formation f après initialisation
 */

    f->nbUE = 0;
    f->nbEtudiant = 0;
    for (int l = 0; l < MAX_UE; ++l) {
        for (int k = 0; k < MAX_EPREUVES; k++) {
            for (int j = 0; j < MAX_MATIERES; ++j) {
                for (int i = 0; i < NB_SEMESTRES; ++i) {
                    for (int h = 0; h < MAX_ETUDIANTS; ++h) {
                        f->etudiants[h].semestres[i].ma-
tieres[j].epreuves[k].coef[l] = 0;
                        f->etudiants[h].semestres[i].ma-
tieres[j].epreuves[k].note[l] = -1; //Pas de note
                        f->etudiants[h].semestres[i].nbMatiere = 0;
                        f->etudiants[h].semestres[i].moyenneNotes[l] = 0;
                        f->etudiants[h].semestres[i].matieres[j].nbEpreuves
= 0;
                        f->etudiants[h].semestres[i].matieres[j].nbNotes =
0;
                    }
                }
            }
        }
    }
}

```



```

}

void cmdFormation(Formation* f, unsigned int tempUE) {
/**
 * Initialise le nombre d'UE de la formation et affiche les messages corres-
 * pondant à la commande 2
 * @param[in] Structure Formation f
 * @param[in] tempUE le nombre d'UE entrée
 * @param[out] Structure Formation f après initialisation du nombre d'UE
 * @pre tempUE valide, compris entre 3 et 6 inclus
 */

    scanf("%i", &tempUE);

    if (tempUE >= MIN_UE && tempUE <= MAX_UE) {
        if (f->nbUE > 0) printf("Le nombre d'UE est déjà défini\n");
        else {
            f->nbUE = tempUE;
            printf("Le nombre d'UE est défini\n");
        }
    }
    else printf("Le nombre d'UE est incorrect\n");
}

bool verifCoef(const Formation* f, unsigned int nSemestre, unsigned int e,
CH30 cmd) {
/**
 * Verifie si les coefficients sont correctes
 * @param[in] Structure Formation f
 * @param[in] nSemestre correspondant au numéro de semestre entré
 * @param[in] e correspondant au numéro de l'étudiant
 * @param[in] cmd correspondant à la commande entrée
 * @pre nSemestre et e valides, nSemestre doit être égal à 0 ou 1 dans la
 * partie decision et 1 ou 2 dans les autres parties, e doit être compris entre
 * 0 et 99
 * @return true ou false si les coefficients sont correctes
 */

    if (!strcmp(cmd, "decision")) {

        assert(((nSemestre == 0 || nSemestre == 1) && ((e >= 0 && e <
MAX_ETUDIANTS))));

        //Somme des coefficients
        //Table où on met la somme des coefficients par UE
        float sumUE[MAX_UE] = { 0,0,0,0,0,0 };
        //3 boucles parcourant tous les UE, toutes les matières et toutes
les epreuves
        for (unsigned int k = 0; k < f->nbUE; ++k) {
            float sumUEpEP = 0; //Après avoir fait la somme d'une UE, on la
reinitialise à 0

```

```

        for (unsigned int j = 0; j < f->etudiants[e].se-
mestres[nSemestre].nbMatières; ++j) {
            for (unsigned int i = 0; i < f->etudiants[e].se-
mestres[nSemestre].matieres[j].nbEpreuves; ++i) {
                //Faire la somme des coefficients par UE, par matières,
par épreuves
                sumUEpEP = sumUEpEP + f->etudiants[e].se-
mestres[nSemestre].matieres[j].epreuves[i].coef[k];
            } //Fin boucle parcourant toutes les épreuves
        } //Fin boucle parcourant toutes les matières
        sumUE[k] = sumUEpEP;
    } // Fin boucle parcourant tous les UE

    //Vérification si tous les coefficients entrés sont null
    float vsumUE = 0; //valeur non null de la table sumUE
    for (unsigned int i = 0; i < f->nbUE; ++i) if (sumUE[i] != 0) vsu-
mUE++;
    if (vsumUE == f->nbUE) return true;
    else return false;

}
else {

    assert(((nSemestre == 1 || nSemestre == 2)) && ((e >= 0 && e <
MAX_ETUDIANTS)));

    //Somme des coefficients
    float sumUE[MAX_UE] = { 0,0,0,0,0,0 };
    for (unsigned int k = 0; k < f->nbUE; ++k) {
        float sumUEpEP = 0;
        for (unsigned int j = 0; j < f->etudiants[e].semetres[nSemestre
- 1].nbMatières; ++j) {
            for (unsigned int i = 0; i < f->etudiants[e].semetres[nSe-
mestre - 1].matieres[j].nbEpreuves; ++i) {
                sumUEpEP = sumUEpEP + f->etudiants[e].se-
mestres[nSemestre - 1].matieres[j].epreuves[i].coef[k];
            }
        }
        sumUE[k] = sumUEpEP;
    }

    //Vérification si tous les coefficients entrés sont null
    float vsumUE = 0; //valeur non null de la table sumUE
    for (unsigned int i = 0; i < f->nbUE; ++i) if (sumUE[i] != 0) vsu-
mUE++;
    if (vsumUE == f->nbUE) return true;
    else return false;

}

}

unsigned int maxStringMat(const Formation* f) {

```

```

/**
 * Trouve la matière qui a la plus longue chaîne de caractère
 * @param[in] Structure Formation f
 * @return max la plus longue chaîne de caractère des matières
 */

    unsigned int tab[MAX_MATIERES], max = 0;

    for (unsigned int i = 0; i < MAX_MATIERES; ++i) tab[i] = 0; //init
    for (unsigned int i = 0; i < f->etudiants[0].semestres[0].nbMatières; ++i)
tab[i] = strlen(f->etudiants[0].semestres[0].matieres[i].nom); //recherche
    for (unsigned int i = 0; i < MAX_MATIERES; ++i) if (tab[i] > max) max =
tab[i]; //recherche le max

    return max;
}

void afficheUE(const Formation* f, CH30 cmd) {
/**
 * Affiche les UE
 * @param[in] Structure Formation f
 * @param[in] cmd correspondant à la commande entrée
 */
    if (!strcmp(cmd, "releve")) {
        for (unsigned int i = 0; i < maxStringMat(*f) + 2; ++i) printf("
"); //Espace
        for (unsigned int i = 0; i < f->nbUE; ++i) {
            if (i == f->nbUE - 1) printf("UE%i \n", i + 1); else
printf("UE%i ", i + 1); //Affiche
        }
    }
    else {
        //Pour commande decision
        for (unsigned int i = 0; i < strlen("Moyennes annuelles") + 2; ++i)
printf(" ");
        for (unsigned int i = 0; i < f->nbUE; ++i) {
            if (i == f->nbUE - 1) printf("UE%i \n", i + 1); else
printf("UE%i ", i + 1);
        }
    }
}

void afficheNomMatAvecMoyenne(const Formation* f, unsigned int nSemestre,
unsigned int e) {
/**
 * Affiche le nom de la matière et ses moyennes par UE
 * @param[in] Structure Formation f
 * @param[in] nSemestre correspondant au numéro de semestre entré
 * @pre nSemestre et e valides, nSemestre doit être égal à 1 ou 2, e doit
être compris entre 0 et 99
 * @param[in] e correspondant au numéro de l'étudiant
 */
}

```

```

    assert(((nSemestre == 1 || nSemestre == 2)) && ((e >= 0 && e < MAX_ETU-
DIANTS))));

    //Affiche le nom de la matière
    for (unsigned int k = 0; k < f->etudiants[e].semestres[nSemestre -
1].nbMatières; ++k) {
        printf("%s", f->etudiants[e].semestres[nSemestre - 1].ma-
tières[k].nom);
        for (unsigned int j = 0; j < (maxStringMat(&f) + 2) - strlen(f-
>etudiants[e].semestres[nSemestre - 1].matières[k].nom) - 1; ++j) printf("
");

        //Affiche les moyennes des matières par UE
        float sumNoteMatT[MAX_UE] = { 0,0,0,0,0,0 };
        float sumCoeffMatT[MAX_UE] = { 0,0,0,0,0,0 };
        float moyenneNoteMatT[MAX_UE] = { 0,0,0,0,0,0 };
        for (unsigned int i = 0; i < f->nbUE; ++i) {
            float sumNoteMat = 0;
            float sumCoeffMat = 0;
            for (unsigned int j = 0; j < f->etudiants[e].semestres[nSemestre
- 1].matières[k].nbEpreuves; ++j) { //Epreuve
                sumNoteMat = sumNoteMat + f->etudiants[e].semestres[nSemes-
tre - 1].matières[k].epreuves[j].note[i] * f->etudiants[e].semestres[nSemes-
tre - 1].matières[k].epreuves[j].coef[i];
                sumCoeffMat = sumCoeffMat + f->etudiants[e].se-
mestres[nSemestre - 1].matières[k].epreuves[j].coef[i];
            }
            sumNoteMatT[i] = sumNoteMat;
            sumCoeffMatT[i] = sumCoeffMat;

            if (sumCoeffMatT[i] == 0) {
                if (i == f->nbUE - 1) printf("  ND \n");
                else printf("  ND ");
            }
            else {

                moyenneNoteMatT[i] = (sumNoteMatT[i] / sumCoeffMatT[i]);

                if (i == f->nbUE - 1) {
                    if (moyenneNoteMatT[i] < 10) printf(" %.1f \n",
floorf(moyenneNoteMatT[i] * 10.f) / 10.f); //Pour arrondir à son entier na-
turel inférieur
                    else printf("%.1f \n", floorf(moyenneNoteMatT[i] * 10.f)
/ 10.f);
                }
                else {
                    if (moyenneNoteMatT[i] < 10) printf(" %.1f ",
floorf(moyenneNoteMatT[i] * 10.f) / 10.f);
                    else printf("%.1f ", floorf(moyenneNoteMatT[i] * 10.f) /
10.f);
                }
            }
        }
    }
}

```

```

    }

    }

}

void afficheMoyenneUEparSemestre(Formation* f, unsigned int nSemestre, unsigned int e, CH30 cmd) {
/**
 * Affiche les moyennes par UE du semestre
 * @param[in] Structure Formation f
 * @param[in] nSemestre correspondant au numéro de semestre entré
 * @param[in] e correspondant au numéro de l'étudiant
 * @pre nSemestre et e valides, nSemestre doit être égal à 1 ou 2 dans la
partie releve et 0 ou 1 dans la partie decision, e doit être compris entre 0
et 99
 * @param[in] cmd correspondant à la commande entrée
 */

    if (!strcmp(cmd, "releve")) {

        assert(((nSemestre == 1 || nSemestre == 2)) && ((e >= 0 && e <
MAX_ETUDIANTS)));

        //Moyenne par UE

        printf("Moyennes"); for (unsigned int i = 0; i < (maxStringMat(*&f)
+ 2) - strlen("Moyennes") - 1; ++i) printf(" ");

        float sumNoteT[MAX_UE] = { 0,0,0,0,0,0 };
        float sumCoeffT[MAX_UE] = { 0,0,0,0,0,0 };
        float moyenneNoteUET[MAX_UE] = { 0,0,0,0,0,0 };
        for (unsigned int i = 0; i < f->nbUE; ++i) {
            float sumNote = 0;
            float sumCoeff = 0;
            float moyenneNoteUE = 0;
            for (unsigned int k = 0; k < f->etudiants[e].semestres[nSemestre
- 1].nbMatières; ++k) {
                for (unsigned int j = 0; j < f->etudiants[e].semestres[nSe-
mestre - 1].matieres[k].nbEpreuves; ++j) {
                    sumNote = sumNote + f->etudiants[e].semestres[nSemestre
- 1].matieres[k].epreuves[j].note[i] * f->etudiants[e].semestres[nSemestre -
1].matieres[k].epreuves[j].coef[i];
                    sumCoeff = sumCoeff + f->etudiants[e].se-
mestres[nSemestre - 1].matieres[k].epreuves[j].coef[i];
                }

            }

            sumNoteT[i] = sumNote;
            sumCoeffT[i] = sumCoeff;
            moyenneNoteUET[i] = (sumNoteT[i] / sumCoeffT[i]);

```

```

        if (i == f->nbUE - 1) {
            if (moyenneNoteUET[i] < 10) printf(" %.1f \n", floorf(moy-
enneNoteUET[i] * 10.f) / 10.f);
            else printf("%.1f \n", floorf(moyenneNoteUET[i] * 10.f) /
10.f);
        }
        else {
            if (moyenneNoteUET[i] < 10) printf(" %.1f ", floorf(moy-
enneNoteUET[i] * 10.f) / 10.f);
            else printf("%.1f ", floorf(moyenneNoteUET[i] * 10.f) /
10.f);
        }
    }
    }
    else {

        assert(((nSemestre == 0 || nSemestre == 1)) && ((e >= 0 && e <
MAX_ETUDIANTS)));

        //Moyenne par UE d'un Semestre

        printf("S%i", nSemestre + 1); for (unsigned int i = 0; i <
(strlen("Moyennes annuelles") + 1) - strlen("Sn"); ++i) printf(" ");

        float sumNoteT[MAX_UE] = { 0,0,0,0,0,0 };
        float sumCoeffT[MAX_UE] = { 0,0,0,0,0,0 };
        float moyenneNoteUET[MAX_UE] = { 0,0,0,0,0,0 }; //Tableau contenant
la moyenne des UE par Semestre
        for (unsigned int i = 0; i < f->nbUE; ++i) {
            float sumNote = 0;
            float sumCoeff = 0;
            float moyenneNoteUE = 0;
            for (unsigned int k = 0; k < f->etudiants[e].semestres[nSemes-
tre].nbMatières; ++k) {
                for (unsigned int j = 0; j < f->etudiants[e].semestres[nSe-
mestre].matieres[k].nbEpreuves; ++j) {
                    sumNote = sumNote + f->etudiants[e].se-
mestres[nSemestre].matieres[k].epreuves[j].note[i] * f->etudiants[e].se-
mestres[nSemestre].matieres[k].epreuves[j].coef[i];
                    sumCoeff = sumCoeff + f->etudiants[e].se-
mestres[nSemestre].matieres[k].epreuves[j].coef[i];
                }

            }
            sumNoteT[i] = sumNote;
            sumCoeffT[i] = sumCoeff;
            moyenneNoteUET[i] = (sumNoteT[i] / sumCoeffT[i]);
            f->etudiants[e].semestres[nSemestre].moyenneNotes[i] = moyenne-
NoteUET[i];

```

```

        if (i == f->nbUE - 1) {
            if (moyenneNoteUET[i] < 10) printf(" %.1f \n", floorf(moy-
enneNoteUET[i] * 10.f) / 10.f);
            else printf("%.1f \n", floorf(moyenneNoteUET[i] * 10.f) /
10.f);
        }
        else {
            if (moyenneNoteUET[i] < 10) printf(" %.1f ", floorf(moy-
enneNoteUET[i] * 10.f) / 10.f);
            else printf("%.1f ", floorf(moyenneNoteUET[i] * 10.f) /
10.f);
        }
    }
}
}

void afficheMoyenneAnnuelleAcqDev(const Formation *f, unsigned int e) {
/**
 * Calcul et affichage des moyennes annuelles d'un étudiant par UE, des UE
acquises, de la decision
 * @param[in] Structure Formation f
 * @param[in] e correspondant au numéro de l'étudiant
 * @pre e valide, compris entre 0 et 99
 */

    assert(e >= 0 && e < MAX_ETUDIANTS);

    unsigned int moyP = 0, temp = 0; //Compteur de moyenne au dessus de la
moyenne (>=10)

    //Moyenne annuelle
    float moyenneNoteSTab[MAX_UE] = { 0,0,0,0,0,0 }; //Tableau contenant la
moyenne annuelle des Semestres par UE
    printf("Moyennes annuelles"); for (unsigned int i = 0; i <
(strlen("Moyennes annuelles") + 1) - strlen("Moyennes annuelles"); ++i)
printf(" ");

    for (unsigned int i = 0; i < f->nbUE; ++i) {
        float sumMoyenne = 0, comptSem = 0, moyenneNoteUE = 0;
        for (unsigned int k = 0; k < NB_SEMESTRES; ++k) {
            sumMoyenne = sumMoyenne + f->etudiants[e].semestres[k].moyenne-
NoteS[i];
            ++comptSem;
        }
        moyenneNoteSTab[i] = (sumMoyenne / comptSem);
        if (i == f->nbUE - 1) {
            if (moyenneNoteSTab[i] < 10) printf(" %.1f \n", floorf(moy-
enneNoteSTab[i] * 10.f) / 10.f);
            else printf("%.1f \n", floorf(moyenneNoteSTab[i] * 10.f) /
10.f);
        }
        else {

```

```

        if (moyenneNoteSTab[i] < 10) printf(" %.1f ", floorf(moy-
enneNoteSTab[i] * 10.f) / 10.f);
        else printf("%.1f ", floorf(moyenneNoteSTab[i] * 10.f) / 10.f);
    }
}

//Acquisition
printf("Acquisition"); for (unsigned int i = 0; i < (strlen("Moyennes
annuelles")) - (strlen("Acquisition") - 1); ++i) printf(" ");

for (unsigned int i = 0; i < f->nbUE; ++i) {
    if (moyenneNoteSTab[i] >= 10) {
        ++moyP;
        if (i == 0 || moyP == 1) printf("UE%i", i + 1);
        else {
            if (i == f->nbUE - 1) {
                printf(", UE%i\n", i + 1);
                ++temp;
            }
            else printf(", UE%i", i + 1);
        }
    }
}

if (moyP == 0) {
    printf("Aucune\n");
    ++temp;
}

if (temp == 0) printf("\n");

//Devenir
printf("Devenir"); for (unsigned int i = 0; i < (strlen("Moyennes an-
nuelles")) - (strlen("Devenir") - 1); ++i) printf(" ");
if (moyP > f->nbUE/2) printf("Passage\n");
else printf("Redoublement\n");
}

int main() {

    //Declaration de structure
    Formation f;

    //Initialisation des valeurs à initialiser
    init(&f);

    //Declarations de variables
    CH30 cmd = "";
    unsigned int tempUE = 0;
    unsigned int nSemestre = 0;
    CH30 nomMatiere;
    CH30 nomEpreuve;

```



```

float coeffUE[MAX_UE] = { 0,0,0,0,0,0 };
CH30 nomEtudiant;
float note;
unsigned int nbEtuTemp = 0;

while (1) {
    scanf("%s", cmd);

    //Commande 1
    if (!strcmp(cmd, "exit")) {
        return 0;
    } //Fin C1

    //Commande 2
    else if (!strcmp(cmd, "formation")) {
        //Initialise le nombre d'UE de la formation
        cmdFormation(&f, tempUE);
    } // Fin C2

    //Commande 3
    else if (!strcmp(cmd, "epreuve")) {

        //Ajoute des epreuves
        //Precondition : nSemestre doit être egal à 1 ou 2

        //Lecture parametres
        scanf("%i %s %s", &nSemestre, nomMatiere, nomEpreuve);
        for (unsigned int i = 0; i < f.nbUE; ++i) scanf("%f", &coef-
fUE[i]);

        if (f.nbUE == 0) printf("Le nombre d'UE n'est pas defini\n");
        else {
            //Verification si le numero de semestre entré est correct
            if (nSemestre == 1 || nSemestre == 2) {

                //Verification si tout les coefficients entrés sont null
                unsigned int vCoefftab = 0; //valeur non null de la
table coeff

                for (unsigned int i = 0; i < f.nbUE; ++i) {
                    if (coeffUE[i] < 0) {
                        vCoefftab = -1;
                        break;
                    }
                    else if (coeffUE[i] > 0) vCoefftab++;
                }

                if (vCoefftab == -1) printf("Au moins un des coeffi-
cients est incorrect\n");
                else if (vCoefftab >= 0) {
                    for (unsigned int e = 0; e < MAX_ETUDIANTS; ++e) {

```

```

        ++nbEtuTemp;
        unsigned int m = 0, n = 0;
        //Recherche matiere connu
        while (m < f.etudiants[e].semestres[nSemestre -
1].nbMatiere && strcmp(nomMatiere, f.etudiants[e].semestres[nSemestre -
1].matieres[m].nom) != 0) ++m;
        if (m < f.etudiants[e].semestres[nSemestre -
1].nbMatiere) { //trouvé

                //Recherche epreuve connu
                while (n < f.etudiants[e].se-
mestres[nSemestre - 1].matieres[m].nbEpreuves && strcmp(nomEpreuve, f.etu-
diants[e].semestres[nSemestre - 1].matieres[m].epreuves[n].nom) != 0) ++n;
                if (n < f.etudiants[e].semestres[nSemestre -
1].matieres[m].nbEpreuves) { //trouvé
                        if (nbEtuTemp > 1); //Test pour afficher
le message qu'une fois

                                else printf("Une meme epreuve existe
deja\n");

                                }
                                else { // pas trouvé, on mémorise l'epreuve
et les coeff

                                        if (vCoefftab == 0) {
                                                if (nbEtuTemp > 1); //Test pour af-
ficher le message qu'une fois

                                                        else printf("Au moins un des coeffi-
cients est incorrect\n");
                                        }
                                        else {
                                                strcpy(f.etudiants[e].se-
mestres[nSemestre - 1].matieres[m].epreuves[n].nom, nomEpreuve);
                                                ++f.etudiants[e].semestres[nSemestre
- 1].matieres[m].nbEpreuves;

                                                        if (nbEtuTemp > 1); //Test pour af-
ficher le message qu'une fois

                                                        else printf("Epreuve ajoutée a la
formation\n");

                                                        //memorise les coeff
                                                        for (unsigned int i = 0; i < f.nbUE;
++i) f.etudiants[e].semestres[nSemestre - 1].matieres[m].epreuves[n].coef[i]
= coeffUE[i];
                                                        }
                                                }
                                        }
                                }
                                else { //matiere pas connu
                                        if (vCoefftab == 0) {
                                                if (nbEtuTemp > 1); //Test pour afficher
le message qu'une fois

```

```

else printf("Au moins un des coeffi-
cients est incorrect\n");
    }
    else {
        //memorise matiere
        strcpy(f.etudiants[e].se-
mestres[nSemestre - 1].matieres[m].nom, nomMatiere);
        ++f.etudiants[e].semestres[nSemestre -
1].nbMatieres;

        if (nbEtuTemp > 1); //Test pour afficher
le message qu'une fois

        else printf("Matiere ajoutee a la forma-
tion\n");

        //memorise l'epreuve d'une matiere in-
connu

        strcpy(f.etudiants[e].se-
mestres[nSemestre - 1].matieres[m].epreuves[n].nom, nomEpreuve);
        ++f.etudiants[e].semestres[nSemestre -
1].matieres[m].nbEpreuves;

        if (nbEtuTemp > 1); //Test pour afficher
le message qu'une fois

        else printf("Epreuve ajoutee a la forma-
tion\n");

        //memorise les coeff
        for (unsigned int i = 0; i < f.nbUE;
++i) f.etudiants[e].semestres[nSemestre - 1].matieres[m].epreuves[n].coef[i]
= coeffUE[i];
    }
}
}
} //Fin boucle etudiant
nbEtuTemp = 0;
}
}
else printf("Le numero de semestre est incorrect\n");
} //Fin test UE existe
} //Fin C3

//Commande 4
else if (!strcmp(cmd, "coefficients")) {

    //Verifie les coefficients
    //Precondition : nSemestre doit être egal à 1 ou 2

    scanf("%i", &nSemestre);

    if (f.nbUE == 0) printf("Le nombre d'UE n'est pas defini\n");
    else {
        //Test verification numero de semestre

```

```

        if (nSemestre == 1 || nSemestre == 2) {
            for (unsigned int e = 0; e < MAX_ETUDIANTS; ++e) {
                ++nbEtuTemp;
                //Test si le semestre contient au moins une epreuve
                if (f.etudiants[e].semestres[nSemestre - 1].nbMatieres != 0) {
                    if (verifCoef(&f, nSemestre, e, cmd)) {
                        if (nbEtuTemp > 1); //Test pour afficher le
message qu'une fois
                        else printf("Coefficients corrects\n");
                    }
                    else {
                        if (nbEtuTemp > 1); //Test pour afficher le
message qu'une fois
                        else printf("Les coefficients d'au moins une
UE de ce semestre sont tous nuls\n");
                    }
                }
                else {
                    if (nbEtuTemp > 1); //Test pour afficher le mes-
sage qu'une fois
                    else printf("Le semestre ne contient aucune
epreuve\n");
                }
            } //Fin Test contient epreuve
        } //Fin boucle etu
        nbEtuTemp = 0;
    } //Fin Test semestre
    else printf("Le numero de semestre est incorrect\n");
} //Fin test UE existe

} //Fin C4

//Commande 5
else if (!strcmp(cmd, "note")) {

    //Ajoute les notes de l'étudiant
    //Precondition : nSemestre doit être égal à 1 ou 2, note doit
être compris entre 0 et 20 inclus, nomMatiere et nomEpreuve doivent être
connus

    scanf("%i %s %s %s %f", &nSemestre, nomEtudiant, nomMatiere,
nomEpreuve, &note);

    if (f.nbUE == 0) printf("Le nombre d'UE n'est pas defini\n");
    else {
        //Verification si le numero de semestre entré est correct
        if (nSemestre == 1 || nSemestre == 2) {

            //Declaration de variable temporaire
            unsigned int m = 0, ep = 0, e = 0, n = 0;

```

```

        //Recherche matiere connue
        while (m < f.etudiants[e].semestres[nSemestre - 1].nbMatieres && strcmp(nomMatiere, f.etudiants[e].semestres[nSemestre - 1].matieres[m].nom) != 0) ++m;
        if (m < f.etudiants[e].semestres[nSemestre - 1].nbMatieres) { //trouvé

            //Recherche epreuve connue
            while (ep < f.etudiants[e].semestres[nSemestre - 1].matieres[m].nbEpreuves && strcmp(nomEpreuve, f.etudiants[e].semestres[nSemestre - 1].matieres[m].epreuves[ep].nom) != 0) ++ep;
            if (ep < f.etudiants[e].semestres[nSemestre - 1].matieres[m].nbEpreuves) { //trouvé

                //Verification si la note entrée est correct
                if (note >= MIN_NOTE && note <= MAX_NOTE) {
                    //Recherche etudiant connu
                    while (e < f.nbEtudiant && strcmp(nomEtudiant, f.etudiants[e].nom) != 0) ++e;
                    if (e < f.nbEtudiant) { //trouvé

                        //recherche note connu
                        if (f.etudiants[e].semestres[nSemestre - 1].matieres[m].epreuves[ep].note[n] > -1) { //trouvé
                            printf("Une note est deja definie pour cet etudiant\n");
                        }
                        else {
                            for (; n < f.nbUE; ++n) f.etudiants[e].semestres[nSemestre - 1].matieres[m].epreuves[ep].note[n] = note;
                            ++f.etudiants[e].semestres[nSemestre - 1].matieres[m].nbNotes;
                            printf("Note ajoutee a l'etudiant\n");
                        }
                    }
                    else { // pas trouvé, on mémorise l'etudiant et sa note
                        strcpy(f.etudiants[e].nom, nomEtudiant);
                        ++f.nbEtudiant;
                        printf("Etudiant ajoute a la formation\n");

                        for (; n < f.nbUE; ++n) f.etudiants[e].semestres[nSemestre - 1].matieres[m].epreuves[ep].note[n] = note;
                        ++f.etudiants[e].semestres[nSemestre - 1].matieres[m].nbNotes;
                        printf("Note ajoutee a l'etudiant\n");
                    }
                }
            }
        }
    }
}

```

```

        else printf("Note incorrecte\n");
    }
    else printf("Epreuve inconnue\n");
}
else printf("Matiere inconnue\n");
}
else printf("Le numero de semestre est incorrect\n");
} //Fin test UE defini

} //Fin C5

//Commande 6
else if (!strcmp(cmd, "notes")) {

    //Verifie les notes entrée
    //Precondition : nSemestre doit être egal à 1 ou 2, nomEtudiant
    doit être connu

    scanf("%i %s", &nSemestre, nomEtudiant);

    if (f.nbUE == 0) printf("Le nombre d'UE n'est pas defini\n");
    else {
        //Verification si le numero de semestre entrée est correct
        if (nSemestre == 1 || nSemestre == 2) {
            unsigned int e = 0, n = 0;
            //Recherche etudiant connu
            while (e < f.nbEtudiant && strcmp(nomEtudiant, f.etu-
dians[e].nom) != 0) ++e;
            if (e < f.nbEtudiant) { //trouvé
                for (unsigned int m = 0; m < MAX_MATIERES; ++m) {
                    if (f.etudiants[e].semestres[nSemestre - 1].ma-
tieres[m].nbNotes == f.etudiants[e].semestres[nSemestre - 1].ma-
tieres[m].nbEpreuves) n++;
                }
                else n--;
            }
            if (n == 10) printf("Notes correctes\n");
            else printf("Il manque au moins une note pour cet
etudiant\n");
        }
        else printf("Etudiant inconnu\n"); // pas trouvé
    }
    else printf("Le numero de semestre est incorrect\n");
} //Fin test UE defini

} //Fin C6

//Commande 7
else if (!strcmp(cmd, "releve")) {

    //Affichage du relevé

```

```

        //Precondition : nSemestre doit être egal à 1 ou 2, nomEtudiant
doit être connu

scanf("%i %s", &nSemestre, nomEtudiant);

if (f.nbUE == 0) printf("Le nombre d'UE n'est pas defini\n");
else {
    //Verification si le numero de semestre entré est correct
    if (nSemestre == 1 || nSemestre == 2) {

        unsigned int e = 0, n = 0;
        //Recherche etudiant connu
        while (e < f.nbEtudiant && strcmp(nomEtudiant, f.etu-
diants[e].nom) != 0) ++e;
        if (e < f.nbEtudiant) { //trouvé

            if (verifCoef(&f, nSemestre, e, cmd)) {
                for (unsigned int m = 0; m < MAX_MATIERES; ++m)
                {
                    if (f.etudiants[e].semestres[nSemestre -
1].matieres[m].nbNotes == f.etudiants[e].semestres[nSemestre - 1].ma-
tieres[m].nbEpreuves) n++;
                    else n--;
                }
                if (n == 10) {
                    //Affichage du relevé
                    //Affiche les UE
                    afficheUE(&f, cmd);

                    //Affiche le nom de la matière et ses
moyennes par UE
                    afficheNomMatAvecMoyenne(&f, nSemestre, e);
                    printf("--\n");

                    //Affiche les moyennes par UE
                    afficheMoyenneUEparSemestre(&f, nSemestre,
e, cmd);
                }
                else printf("Il manque au moins une note pour
cet etudiant\n");
            }
            else printf("Les coefficients de ce semestre sont
incorrects\n");
        }
        else printf("Etudiant inconnu\n");
    }
    else printf("Le numero de semestre est incorrect\n");
} //Fin test UE defini

} //Fin C7

```

```

//Commande 8
else if (!strcmp(cmd, "decision")) {

    //Affiche le relevé des notes annuelles avec les UE acquises et
    la decision du jury
    //Precondition : nSemestre doit être egal à 1 ou 2, nomEtudiant
    doit être connu

    scanf("%s", nomEtudiant);
    int nbrEtuTemp = 0;
    if (f.nbUE == 0) printf("Le nombre d'UE n'est pas defini\n");
    else {
        unsigned int e = 0, n = 0;
        //Recherche etudiant connu
        while (e < f.nbEtudiant && strcmp(nomEtudiant, f.etu-
dians[e].nom) != 0) ++e;
        if (e < f.nbEtudiant) { //trouvé
            for (unsigned int s = 0; s < NB_SEMESTRES; ++s) {
                if (nbrEtuTemp != 0);
                else {
                    if (verifCoef(&f, s, e, cmd)) {
                        for (unsigned int s = 0; s < NB_SEMESTRES;
++s) {
                            for (unsigned int m = 0; m <
MAX_MATIERES; ++m) {
                                if (f.etudiants[e].semes-
tres[s].matieres[m].nbNotes == f.etudiants[e].semes-
tres[s].matieres[m].nbEpreuves) n++;
                                else n--;
                            }
                        }
                    }
                    if (n == 20) {

                        //Affichage du relevé et decision

                        //Affiche les UE
                        afficheUE(&f, cmd);

                        //Affiche les moyennes par UE des
semestres

                        for (unsigned int sem = 0; sem <
NB_SEMESTRES; ++sem) afficheMoyenneUEparSemestre(&f, sem, e, cmd);
                        printf("--\n");

                        //Affiche les moyennes annuelles de
chaque UE, les acquisitions et la decision
                        afficheMoyenneAnnuelleAcqDev(&f, e);

                    }
                    else printf("Il manque au moins une note
pour cet etudiant\n");
                }
            }
        }
    }
}

```



```

        }
        else printf("Les coefficients d'au moins un se-
mestre sont incorrects\n");
    }
    ++nbrEtuTemp;
}
}
else printf("Etudiant inconnu\n");
} //Fin test UE defini

} //Fin C8

} //Fin de la boucle infini

} //Fin du main

```

- Trace d'exécution du test du sprint le plus haut

Pendant la recette, nous avons validé le sprint 4 avec les alignements et le sprint 5 sans les alignements.

Ci-dessous, vous retrouverez le test du sprint le plus haut :

SPRINT 4 BASE	
IN	OUT
formation 3	Le nombre d'UE est defini
epreuve 1 Programmation Projet 1 2 0	Matiere ajoutee a la formation
epreuve 1 Programmation DST 2 3 0	Epreuve ajoutee a la formation
epreuve 1 SGBD Participation 0.5 0 0.5	Epreuve ajoutee a la formation
epreuve 1 SGBD Rapport 1.5 0 1.5	Matiere ajoutee a la formation
epreuve 2 Architecture Interrogation 1 0 2	Epreuve ajoutee a la formation
epreuve 2 Architecture DST 0 1 4	Epreuve ajoutee a la formation
epreuve 2 Systeme QCM 2 3 0.5	Matiere ajoutee a la formation
epreuve 2 Systeme Expose 3 2 0.5	Epreuve ajoutee a la formation
note 1 Paul Programmation Projet 12	Epreuve ajoutee a la formation
note 1 Paul Programmation DST 9	Matiere ajoutee a la formation
note 1 Paul SGBD Participation 16	Epreuve ajoutee a la formation
note 1 Paul SGBD Rapport 12	Epreuve ajoutee a la formation
note 2 Paul Architecture Interrogation 18	Etudiant ajoute a la formation
note 2 Paul Architecture DST 12	Note ajoutee a l'etudiant
note 2 Paul Systeme QCM 7	Note ajoutee a l'etudiant
note 2 Paul Systeme Expose 8	Note ajoutee a l'etudiant
note 1 Paule Programmation Projet 8	Note ajoutee a l'etudiant
note 1 Paule Programmation DST 11	Note ajoutee a l'etudiant
note 1 Paule SGBD Participation 20	Note ajoutee a l'etudiant
note 1 Paule SGBD Rapport 0	Note ajoutee a l'etudiant
note 2 Paule Architecture Interrogation 18	Note ajoutee a l'etudiant
note 2 Paule Architecture DST 12	Etudiant ajoute a la formation
note 2 Paule Systeme QCM 7	Note ajoutee a l'etudiant
note 2 Paule Systeme Expose 8	Note ajoutee a l'etudiant
note 1 Paulo Programmation Projet 12	Note ajoutee a l'etudiant
note 1 Paulo Programmation DST 9	Note ajoutee a l'etudiant
note 1 Paulo SGBD Participation 16	Note ajoutee a l'etudiant
note 1 Paulo SGBD Rapport 12	Note ajoutee a l'etudiant
note 2 Paulo Architecture Interrogation 17	Note ajoutee a l'etudiant
note 2 Paulo Architecture DST 15	Note ajoutee a l'etudiant
note 2 Paulo Systeme QCM 16	Etudiant ajoute a la formation
note 2 Paulo Systeme Expose 19	Note ajoutee a l'etudiant
decision Paul	Note ajoutee a l'etudiant
decision Paule	Note ajoutee a l'etudiant
decision Paulo	Note ajoutee a l'etudiant
exit	Note ajoutee a l'etudiant
	Note ajoutee a l'etudiant
	Note ajoutee a l'etudiant

	<p>UE1 UE2 UE3</p> <p>S1 11.2 10.2 13.0</p> <p>S2 9.3 8.1 13.0</p> <p>--</p> <p>Moyennes annuelles 10.2 9.1 13.0</p> <p>Acquisition UE1, UE3</p> <p>Devenir Passage</p> <p>UE1 UE2 UE3</p> <p>S1 8.0 9.8 5.0</p> <p>S2 9.3 8.1 13.0</p> <p>--</p> <p>Moyennes annuelles 8.6 8.9 9.0</p> <p>Acquisition Aucune</p> <p>Devenir Redoublement</p> <p>UE1 UE2 UE3</p> <p>S1 11.2 10.2 13.0</p> <p>S2 17.6 16.8 15.9</p> <p>--</p> <p>Moyennes annuelles 14.4 13.5 14.4</p> <p>Acquisition UE1, UE2, UE3</p> <p>Devenir Passage</p>
SPRINT 4 ERREUR	
IN	OUT
<p>formation 3</p> <p>epreuve 1 Programmation Projet 1 2 0</p> <p>epreuve 1 Programmation DST 2 3 0</p> <p>epreuve 2 Architecture Interrogation 1 0 2</p> <p>epreuve 2 Architecture DST 0 1 4</p> <p>note 1 Paul Programmation Projet 12</p> <p>note 1 Paul Programmation DST 9</p> <p>note 2 Paul Architecture Interrogation 18</p> <p>note 2 Paul Architecture DST 12</p> <p>decision Paul</p> <p>epreuve 1 SGBD Participation 0.5 0 0.5</p> <p>epreuve 1 SGBD Rapport 1.5 0 1.5</p> <p>note 1 Paul SGBD Participation 16</p> <p>decision Paul</p> <p>note 1 Paul SGBD Rapport 12</p> <p>decision Paulo</p> <p>decision Paul</p> <p>exit</p>	<p>Le nombre d'UE est defini</p> <p>Matiere ajoutee a la formation</p> <p>Epreuve ajoutee a la formation</p> <p>Epreuve ajoutee a la formation</p> <p>Matiere ajoutee a la formation</p> <p>Epreuve ajoutee a la formation</p> <p>Epreuve ajoutee a la formation</p> <p>Etudiant ajoute a la formation</p> <p>Note ajoutee a l'etudiant</p> <p>Note ajoutee a l'etudiant</p> <p>Note ajoutee a l'etudiant</p> <p>Note ajoutee a l'etudiant</p> <p>Les coefficients d'au moins un semestre sont incorrects</p> <p>Matiere ajoutee a la formation</p> <p>Epreuve ajoutee a la formation</p> <p>Epreuve ajoutee a la formation</p> <p>Note ajoutee a l'etudiant</p> <p>Il manque au moins une note pour cet etudiant</p>

	Note ajoutée à l'étudiant
	Étudiant inconnu
	UE1 UE2 UE3
S1	11.2 10.2 13.0
S2	18.0 12.0 14.0
--	
	Moyennes annuelles 14.6 11.1 13.5
	Acquisition UE1, UE2, UE3
	Devenir Passage

- Trace d'exécution d'un de nos jeux de test du sprint le plus haut

Un de nos sprints :

IN	OUT
formation 4	Le nombre d'UE est defini
epreuve 1 anglais oral 1 1 0.5 0	Matiere ajoutee a la formation
epreuve 1 anglais dst 2 2 1 2	Epreuve ajoutee a la formation
epreuve 2 système qcm 1 0.5 0 0	Epreuve ajoutee a la formation
epreuve 1 web dst 3 4 2 0	Matiere ajoutee a la formation
epreuve 3 prog dst 3 4 3 5	Epreuve ajoutee a la formation
epreuve 2 prog dst 3 4 3 5	Matiere ajoutee a la formation
epreuve 2 système dst 1.5 2 0 2	Epreuve ajoutee a la formation
epreuve 2 anglais dst 1.5 2 1 0 0	Le numero de semestre est incorrect
epreuve 2 maths dst 2 1 2 1	Matiere ajoutee a la formation
epreuve 2 anglais oral 1 0 1 0	Epreuve ajoutee a la formation
epreuve 1 anglais participation 0.5 0 0.5 0	Epreuve ajoutee a la formation
epreuve 2 communication dst 1 0 0.5 2	Matiere ajoutee a la formation
epreuve 1 prog projet 4 3 0 4	Epreuve ajoutee a la formation
epreuve 1 communication dst 0.5 1 1 1	Matiere ajoutee a la formation
epreuve 2 gestion projet 3 1 2.5 1	Epreuve ajoutee a la formation
epreuve 1 communication oral 0 1 1 1	Epreuve ajoutee a la formation
epreuve 2 maths interro 3 0.5 1 1	Epreuve ajoutee a la formation
epreuve 2 archi dst 0 2 1 0	Matiere ajoutee a la formation
epreuve 1 système qcm 0 0 1 2	Epreuve ajoutee a la formation
epreuve 4 prog projet 2 4 3 5	Matiere ajoutee a la formation
epreuve 2 prog projet 2 4 3 5	Epreuve ajoutee a la formation
epreuve 2 archi projet 0 2 1 0	Matiere ajoutee a la formation
epreuve 1 système dst 0 0 3 -3	Epreuve ajoutee a la formation
epreuve 1 système dst 0 0 3 3	Matiere ajoutee a la formation
epreuve 2 anglais participation 0 0 0 0	Epreuve ajoutee a la formation
epreuve 2 anglais participation 0 0 0.5 0	Epreuve ajoutee a la formation
epreuve 1 maths dst 3 2.5 1.5 2	Epreuve ajoutee a la formation
epreuve 1 maths interro 1 1.5 0.5 1	Matiere ajoutee a la formation
epreuve 1 web projet 3 0 2 0	Epreuve ajoutee a la formation
epreuve 2 gestion dst 0 1 3 1	Matiere ajoutee a la formation
epreuve 1 prog dst 2 4 6 4	Epreuve ajoutee a la formation
epreuve 2 communication oral 1 0 0.5 2	Le numero de semestre est incorrect
coefficients 1	Epreuve ajoutee a la formation
coefficients 2	Epreuve ajoutee a la formation
coefficients 1	Au moins un des coefficients est incorrect
note 1 jules anglais oral 15	Epreuve ajoutee a la formation
note 1 jules maths dst 12	Au moins un des coefficients est incorrect
note 1 jules web dst 13	Epreuve ajoutee a la formation
note 1 jules anglais participation 13	Matiere ajoutee a la formation
note 1 jules prog dst 10	Epreuve ajoutee a la formation
note 1 jules système qcm 12	Epreuve ajoutee a la formation
note 1 jules prog projet 15	Epreuve ajoutee a la formation
note 1 jules communication oral 14	Epreuve ajoutee a la formation
note 1 jules système dst 13	Epreuve ajoutee a la formation
notes 1 jules	Epreuve ajoutee a la formation
note 1 jules maths interro 11	Coefficients corrects
note 1 jules anglais dst 17	Coefficients corrects
note 1 jules communication dst 15	Coefficients corrects
note 1 jules web projet 18	Etudiant ajoute a la formation

notes 1 jules	Note ajoutée à l'étudiant
note 2 jules prog projet 13	Note ajoutée à l'étudiant
note 2 jules maths dst 9	Note ajoutée à l'étudiant
note 2 jules anglais dst 15	Note ajoutée à l'étudiant
note 2 jules communication dst 12	Note ajoutée à l'étudiant
note 2 jules système qcm 10	Note ajoutée à l'étudiant
note 2 jules maths interro 13	Note ajoutée à l'étudiant
note 2 jules archi projet 14	Note ajoutée à l'étudiant
note 2 jules gestion projet 16	Note ajoutée à l'étudiant
note 2 jules communication oral 9	Il manque au moins une note pour cet étudiant
note 2 jules système dst 18	Note ajoutée à l'étudiant
note 2 jules anglais oral 13	Note ajoutée à l'étudiant
note 2 jules archi dst 13	Note ajoutée à l'étudiant
note 2 jules prog dst 12	Note ajoutée à l'étudiant
note 2 jules anglais participation 17	Notes correctes
note 2 jules gestion dst 15	Note ajoutée à l'étudiant
notes 2 jules	Note ajoutée à l'étudiant
note 1 antoine maths dst 14.5	Note ajoutée à l'étudiant
note 1 antoine web dst 14.5	Note ajoutée à l'étudiant
note 1 antoine anglais participation 14.5	Note ajoutée à l'étudiant
note 1 antoine anglais oral 14.5	Note ajoutée à l'étudiant
note 1 antoine prog projet 14.5	Note ajoutée à l'étudiant
note 2 antoine anglais oral 14.5	Note ajoutée à l'étudiant
note 1 antoine prog dst 15	Note ajoutée à l'étudiant
note 1 antoine système dst 14.5	Note ajoutée à l'étudiant
note 1 antoine système qcm 14.5	Note ajoutée à l'étudiant
note 1 antoine communication oral 16	Note ajoutée à l'étudiant
notes 1 antoine	Note ajoutée à l'étudiant
note 1 antoine maths interro 11	Note ajoutée à l'étudiant
note 1 antoine web projet 14.5	Note ajoutée à l'étudiant
note 1 antoine anglais dst 10.5	Notes correctes
note 1 antoine	Etudiant ajoute à la formation
note 1 antoine communication dst 14.5	Note ajoutée à l'étudiant
notes 1 antoine	Note ajoutée à l'étudiant
note 2 antoine prog projet 14.5	Note ajoutée à l'étudiant
note 2 antoine maths dst 10.5	Note ajoutée à l'étudiant
note 2 antoine gestion dst 14.5	Note ajoutée à l'étudiant
note 2 antoine communication dst 11	Note ajoutée à l'étudiant
note 2 antoine maths interro 14.5	Note ajoutée à l'étudiant
notes 2 antoine	Note ajoutée à l'étudiant
note 2 antoine archi projet 16	Note ajoutée à l'étudiant
note 2 antoine archi dst 14.5	Note ajoutée à l'étudiant
note 2 antoine gestion projet 14.5	Il manque au moins une note pour cet étudiant
note 2 antoine anglais dst 14.5	Note ajoutée à l'étudiant
note 2 antoine communication oral 10.5	Note ajoutée à l'étudiant
note 2 antoine système qcm 13	Note ajoutée à l'étudiant
note 2 antoine système dst 14.5	Matière inconnue
note 2 antoine prog dst 14.5	Il manque au moins une note pour cet étudiant
note 2 antoine anglais participation 10.5	Note ajoutée à l'étudiant
notes 2 antoine	Note ajoutée à l'étudiant
note 1 simon anglais oral 8	Note ajoutée à l'étudiant
note 1 simon communication oral 12	Note ajoutée à l'étudiant
note 1 simon web dst 8	Note ajoutée à l'étudiant
note 1 simon maths interro 11	Il manque au moins une note pour cet étudiant

note 1 simon anglais participation 8	Note ajoutée à l'étudiant
note 1 simon prog dst 12.5	Note ajoutée à l'étudiant
note 1 simon système qcm 8	Note ajoutée à l'étudiant
notes 1 simon	Note ajoutée à l'étudiant
note 1 simon prog projet 3	Note ajoutée à l'étudiant
note 1 simon maths dst 12	Note ajoutée à l'étudiant
note 1 simon web projet 8	Note ajoutée à l'étudiant
note 1 simon système dst 8	Note ajoutée à l'étudiant
note 1 simon anglais dst 10.5	Note ajoutée à l'étudiant
note 1 simon	Notes correctes
note 1 simon communication dst 8	Etudiant ajoute à la formation
note 2 simon prog projet 8	Note ajoutée à l'étudiant
note 2 simon maths dst 10.5	Note ajoutée à l'étudiant
notes 1 simon	Note ajoutée à l'étudiant
note 2 simon anglais dst 15	Note ajoutée à l'étudiant
note 2 simon communication dst 5	Note ajoutée à l'étudiant
note 2 simon archi projet 5	Note ajoutée à l'étudiant
note 2 simon gestion dst 8	Note ajoutée à l'étudiant
note 2 simon anglais oral 19	Il manque au moins une note pour cet étudiant
note 2 simon gestion projet 8	Note ajoutée à l'étudiant
note 2 simon maths interro 8	Note ajoutée à l'étudiant
note 2 simon système dst 8	Note ajoutée à l'étudiant
notes 2 simon	Note ajoutée à l'étudiant
note 2 simon archi dst 6	Note ajoutée à l'étudiant
note 2 simon prog dst 8.5	Matière inconnue
note 2 simon communication oral 10.5	Note ajoutée à l'étudiant
note 2 simon système qcm 8	Note ajoutée à l'étudiant
note 2 simon anglais participation 10.5	Il manque au moins une note pour cet étudiant
notes 2 simon	Note ajoutée à l'étudiant
note 1 emily anglais oral 13.5	Note ajoutée à l'étudiant
note 1 emily anglais dst 7	Note ajoutée à l'étudiant
note 1 emily web dst 13	Note ajoutée à l'étudiant
notes 1 emily	Note ajoutée à l'étudiant
note 2 emily anglais dst 13.5	Note ajoutée à l'étudiant
note 1 emily anglais participation 13	Note ajoutée à l'étudiant
note 1 emily système qcm 9.5	Note ajoutée à l'étudiant
note 1 emily prog projet 13.5	Note ajoutée à l'étudiant
note 2 emily maths dst 9	Il manque au moins une note pour cet étudiant
note 1 emily communication oral 14	Note ajoutée à l'étudiant
note 2 emily anglais oral 13	Note ajoutée à l'étudiant
note 1 emily système dst 13	Note ajoutée à l'étudiant
note 1 emily maths dst 9.5	Note ajoutée à l'étudiant
note 1 emily maths interro 11	Note ajoutée à l'étudiant
note 1 emily communication dst 13.5	Notes correctes
note 1 emily prog dst 9.5	Etudiant ajoute à la formation
note 1 emily web projet 9.5	Note ajoutée à l'étudiant
note 2 emily prog projet 13	Note ajoutée à l'étudiant
note 2 emily système qcm 9.5	Note ajoutée à l'étudiant
note 2 emily maths interro 13	Il manque au moins une note pour cet étudiant
notes 1 emily	Note ajoutée à l'étudiant
note 2 emily archi projet 14	Note ajoutée à l'étudiant
note 2 emily gestion projet 15	Note ajoutée à l'étudiant
note 2 emily communication dst 9.5	Note ajoutée à l'étudiant
notes 2 emily	Note ajoutée à l'étudiant

note 2 emily système dst 9.5	Note ajoutée à l'étudiant
note 2 emily anglais participation 7	Note ajoutée à l'étudiant
note 2 emily archi dst 13	Note ajoutée à l'étudiant
note 2 emily communication oral 9	Note ajoutée à l'étudiant
notes 2 emily	Note ajoutée à l'étudiant
note 2 emily prog dst 9.5	Note ajoutée à l'étudiant
note 2 emily gestion dst 13.5	Note ajoutée à l'étudiant
notes 2 emily	Note ajoutée à l'étudiant
note 1 samir anglais oral 11	Note ajoutée à l'étudiant
note 1 samir prog dst 10	Note ajoutée à l'étudiant
note 1 samir web dst 12	Notes correctes
note 1 samir maths dst 11	Note ajoutée à l'étudiant
note 1 samir anglais participation 12	Note ajoutée à l'étudiant
notes 2 samir	Note ajoutée à l'étudiant
note 1 samir système qcm 11	Il manque au moins une note pour cet étudiant
note 1 samir web projet 17	Note ajoutée à l'étudiant
note 1 samir prog projet 12	Note ajoutée à l'étudiant
note 1 samir système dst 11	Note ajoutée à l'étudiant
notes 1 samir	Note ajoutée à l'étudiant
note 1 samir maths interro 9.5	Il manque au moins une note pour cet étudiant
notes 1 samir	Note ajoutée à l'étudiant
note 1 samir anglais dst 9.5	Note ajoutée à l'étudiant
note 1 samir communication oral 14	Notes correctes
note 1 samir communication dst 11	Etudiant ajoute à la formation
notes 1 samir	Note ajoutée à l'étudiant
note 2 samir prog projet -11	Note ajoutée à l'étudiant
note 2 samir prog projet 11	Note ajoutée à l'étudiant
note 2 samir maths dst 9.5	Note ajoutée à l'étudiant
notes 2 samir	Note ajoutée à l'étudiant
note 2 samir anglais oral 11	Il manque au moins une note pour cet étudiant
note 2 samir archi dst 12	Note ajoutée à l'étudiant
note 2 samir communication dst 12	Note ajoutée à l'étudiant
note 2 samir gestion dst 12	Note ajoutée à l'étudiant
note 2 samir système qcm 10	Note ajoutée à l'étudiant
note 2 samir prog dst 12	Il manque au moins une note pour cet étudiant
note 2 samir anglais participation 9.5	Note ajoutée à l'étudiant
note 2 samir gestion projet 16	Il manque au moins une note pour cet étudiant
note 2 samir anglais dst 11	Note ajoutée à l'étudiant
note 2 samir maths interro 11	Note ajoutée à l'étudiant
note 2 samir archi projet 14	Note ajoutée à l'étudiant
note 2 samir système dst 17	Notes correctes
note 2 samir communication oral -9.5	Note incorrecte
note 2 samir communication oral 9.5	Note ajoutée à l'étudiant
notes 2 samir	Note ajoutée à l'étudiant
releve 2 jules	Il manque au moins une note pour cet étudiant
releve 1 jules	Note ajoutée à l'étudiant
releve 2 emily	Note ajoutée à l'étudiant
releve 2 antoine	Note ajoutée à l'étudiant
releve 1 simon	Note ajoutée à l'étudiant
releve 2 samir	Note ajoutée à l'étudiant
releve 1 samir	Note ajoutée à l'étudiant
releve 1 emily	Note ajoutée à l'étudiant
releve 1 antoine	Note ajoutée à l'étudiant
releve 2 simon	Note ajoutée à l'étudiant

decision simon	Note ajoutee a l'etudiant
decision emily	Note ajoutee a l'etudiant
decision jules	Note ajoutee a l'etudiant
decision antoine	Note incorrecte
decision samir	Note ajoutee a l'etudiant
exit	Notes correctes
	UE1 UE2 UE3 UE4
	système 14.8 16.4 ND 18.0
	prog 12.4 12.5 12.5 12.5
	anglais 14.2 15.0 14.6 ND
	maths 11.4 10.3 10.3 11.0
	communication 10.5 ND 10.5 10.5
	gestion 16.0 15.5 15.4 15.5
	archi ND 13.5 13.5 ND
	--
	Moyennes 13.0 13.5 13.2 12.8
	UE1 UE2 UE3 UE4
	anglais 15.8 16.3 15.5 17.0
	web 15.5 13.0 15.5 ND
	prog 13.3 12.1 10.0 12.5
	communication 15.0 14.5 14.5 14.5
	système ND ND 12.7 12.6
	maths 11.7 11.6 11.7 11.6
	--
	Moyennes 14.1 13.0 12.8 13.0
	UE1 UE2 UE3 UE4
	système 9.5 9.5 ND 9.5
	prog 10.9 11.2 11.2 11.2
	anglais 13.3 13.5 12.0 ND
	maths 11.4 10.3 10.3 11.0
	communication 9.2 ND 9.2 9.2
	gestion 15.0 14.2 14.1 14.2
	archi ND 13.5 13.5 ND
	--
	Moyennes 11.6 11.9 12.1 10.9
	UE1 UE2 UE3 UE4
	système 13.9 14.2 ND 14.5
	prog 14.5 14.5 14.5 14.5
	anglais 14.5 14.5 13.7 ND
	maths 12.9 11.8 11.8 12.5
	communication 10.7 ND 10.7 10.7
	gestion 14.5 14.5 14.5 14.5
	archi ND 15.2 15.2 ND
	--
	Moyennes 13.6 14.4 13.8 13.5
	Il manque au moins une note pour cet etudiant
	UE1 UE2 UE3 UE4
	système 14.2 15.6 ND 17.0
	prog 11.6 11.5 11.5 11.5
	anglais 11.0 11.0 10.7 ND
	maths 10.4 10.0 10.0 10.2
	communication 10.7 ND 10.7 10.7
	gestion 16.0 14.0 13.8 14.0
	archi ND 13.0 13.0 ND

	--
	Moyennes 12.1 12.4 11.9 12.0
	UE1 UE2 UE3 UE4
	anglais 10.2 10.0 10.5 9.5
	web 14.5 12.0 14.5 ND
	prog 11.3 10.8 10.0 11.0
	communication 11.0 12.5 12.5 12.5
	système ND ND 11.0 11.0
	maths 10.6 10.4 10.6 10.5
	--
	Moyennes 11.9 11.0 11.4 10.9
	UE1 UE2 UE3 UE4
	anglais 9.7 9.1 10.1 7.0
	web 11.2 13.0 11.2 ND
	prog 12.1 11.2 9.5 11.5
	communication 13.5 13.7 13.7 13.7
	système ND ND 12.1 11.6
	maths 9.8 10.0 9.8 10.0
	--
	Moyennes 11.0 11.2 10.9 11.0
	Il manque au moins une note pour cet etudiant
	UE1 UE2 UE3 UE4
	système 8.0 8.0 ND 8.0
	prog 8.3 8.2 8.2 8.2
	anglais 16.6 15.0 15.7 ND
	maths 9.0 9.6 9.6 9.2
	communication 7.7 ND 7.7 7.7
	gestion 8.0 8.0 8.0 8.0
	archi ND 5.5 5.5 ND
	--
	Moyennes 9.3 8.4 9.0 8.2
	Il manque au moins une note pour cet etudiant
	UE1 UE2 UE3 UE4
S1	11.0 11.2 10.9 11.0
S2	11.6 11.9 12.1 10.9
	--
	Moyennes annuelles 11.3 11.6 11.5 11.0
	Acquisition UE1, UE2, UE3, UE4
	Devenir Passage
	UE1 UE2 UE3 UE4
S1	14.1 13.0 12.8 13.0
S2	13.0 13.5 13.2 12.8
	--
	Moyennes annuelles 13.5 13.3 13.0 12.9
	Acquisition UE1, UE2, UE3, UE4
	Devenir Passage
	Il manque au moins une note pour cet etudiant
	UE1 UE2 UE3 UE4
S1	11.9 11.0 11.4 10.9
S2	12.1 12.4 11.9 12.0
	--
	Moyennes annuelles 12.0 11.7 11.6 11.4
	Acquisition UE1, UE2, UE3, UE4
	Devenir Passage

