

# BUT1 – SAÈ S1.01

## IMPLÉMENTATION D'UN BESOIN CLIENT

### – GESTION D'UNE FORMATION –

Le but du projet est de développer un interpréteur de commande permettant de gérer une formation universitaire. Les commandes à prendre en compte vont permettre de définir la structure de la formation ainsi que les étudiants qui y sont inscrits, d'affecter des notes à ces étudiants et enfin de réaliser les tâches de fin de semestre et d'année (édition de relevés de notes et établissement des décisions de jury).

## 1 Définitions

Une formation est toujours composée de 2 semestres (désignés respectivement par S1 et S2). Une formation est structurée en unités d'enseignement (UE). Le nombre d'UE est obligatoirement compris entre 3 et 6. À chaque semestre est associé une ou plusieurs matières (les ressources ne sont pas distinguées des SAÈ) et chaque matière prévoit une ou plusieurs épreuves (interrogation, projet, DST, ...). Enfin, à chaque épreuve est associé un coefficient (un réel positif) par UE. L'ensemble de ces informations peut être synthétisé sous la forme d'un tableau comme dans l'exemple ci-dessous.

Semestres	Matières	Épreuves	Coefficients		
			UE1	UE2	UE3
S1	Programmation	Projet	1	2	0
		DST	2	3	0
	SGBD	Participation	0,5	0	0,5
		Rapport	1,5	0	1,5
S2	Architecture	Interrogation	1	0	2
		DST	0	1	4
	Systèmes	QCM	2	3	0,5
		Exposé	3	2	0,5

Table 1: Matières, épreuves et coefficients

Dans l'exemple de la Table 1, la formation est limitée à 3 UE. Chaque semestre est composé de 2 matières et chaque matière prévoit 2 épreuves. On voit que pour une épreuve donnée, le coefficient de certaines UE est égal à 0. Les coefficients peuvent prendre toute valeur. Toutefois, pour une épreuve donnée, au moins un des coefficients d'une UE est nécessairement non nul car toute épreuve doit concourir à la moyenne d'au moins une UE. De même, pour une UE donnée, au moins un des coefficients d'une épreuve de chaque semestre est nécessairement non nul car la moyenne de toute UE doit pouvoir être calculée à chaque semestre.

À l'issu de chaque épreuve, les étudiants reçoivent une note (un réel compris entre 0 et 20). À la fin de chaque semestre, la moyenne de chaque matière est calculée pour chaque UE. Elle correspond à la moyenne pondérée par les coefficients des notes obtenues par l'étudiant aux épreuves de la matière. La moyenne d'UE est elle aussi calculée. Elle correspond à la moyenne des notes des épreuves du semestre. Cette moyenne est pondérée avec les mêmes coefficients. Le tableau ci-dessous présente les notes et les moyennes d'un étudiant donné à l'issu du premier semestre.

Matières	Épreuves	Notes	Moyennes/Matières		
			UE1	UE2	UE3
Programmation	Projet	12	10	10,2	ND
	DST	9			
SGBD	Participation	16	13	ND	13
	Rapport	12			
<b>Moyennes/UE</b>			<b>11,2</b>	<b>10,2</b>	<b>13</b>

Table 2: Notes et moyennes

Dans l'exemple de la Table 2, on voit que l'étudiant a obtenu la note 12 au *Projet* de la matière *Programmation* et 9 au *DST* de cette même matière. La moyenne de cette matière pour l'*UE1* est obtenue en calculant  $((12 \times 1) + (9 \times 2)) \div (1 + 2) = 30 \div 3 = 10$ , c'est à dire en calculant la somme des notes multipliées par leur coefficient respectif et en la

divisant par la somme des coefficients. Les coefficients sont présentés en gras dans le calcul. On remarque que lorsque la somme des coefficients est nulle, la moyenne n'est pas calculée et cela est spécifié par la marque *ND* (pour *Non Définie*). La moyenne d'une UE est obtenue en réalisant le même calcul mais en considérant toutes les épreuves du semestre. Par exemple, la moyenne de l'*UE1* donne  $((12 \times \mathbf{1}) + (9 \times \mathbf{2}) + (16 \times \mathbf{0,5}) + (12 \times \mathbf{1,5})) \div (\mathbf{1} + \mathbf{2} + \mathbf{0,5} + \mathbf{1,5}) = 56 \div 5 = 11,2$ .

À la fin de l'année, les décisions de jury sont prises. Pour ce faire les moyennes de l'année par UE sont calculées. Ce calcul est facile car il correspond à la moyenne simple (non pondérée) des moyennes des deux semestres UE par UE. Les UE dont la moyenne annuelle est supérieure ou égale à 10 sont acquises. L'étudiant passe dans l'année suivante s'il a acquis plus de la moitié des UE de l'année et il redouble dans le cas contraire. Ce calcul est synthétisé dans le tableau suivant.

Semestres	Moyennes/UE		
	UE1	UE2	UE3
S1	11,2	10,2	13
S2	9,3	8,1	13
<b>Moyennes annuelles</b>	<b>10,2</b>	<b>9,1</b>	<b>13</b>
<b>Acquisition</b>	<b>UE1 et UE3</b>		
<b>Devenir</b>	<b>Passage</b>		

Table 3: Moyennes annuelles et décisions

Pour l'étudiant de la Table 3, la moyenne annuelle de l'*UE1* est obtenue par  $(11,2 + 9,3) \div 2 \approx 10,2$ . L'arrondi réalisé ici est expliqué dans la suite de ce document. Les moyennes annuelles de l'*UE1* et l'*UE3* sont supérieures ou égales à 10 et ces UE sont donc acquises alors que l'*UE2* ne l'est pas. L'étudiant a acquis deux UE sur trois et peut donc passer dans l'année suivante de la formation.

## 2 Cahier des charges

L'application que vous devez développer doit interpréter 8 commandes distinctes. Celles-ci sont représentées par des chaînes de caractères de formats spécifiés ci-dessous. Le caractère séparateur entre la commande et les différents champs d'information est le caractère 'espace'.

Les commandes pourront être soit entrées en utilisant l'entrée standard (le clavier), soit par redirection d'un fichier texte sur l'entrée standard (la redirection sera traitée en TP). En tout état de cause, votre programme doit lire les commandes sur l'entrée standard et produire des résultats sur la sortie standard (l'écran). Les commandes sont supposées respectées le format attendu. Seules les erreurs explicitement indiquées dans ce document doivent être prises en compte par votre programme.

Les commandes sont les suivantes :

### C1 - Commande de sortie du programme

Une ligne composée de la chaîne de caractères `exit`.

### C2 - Commande de définition d'une formation

Une ligne composée de la chaîne de caractères `"formation"` suivie du nombre d'UE la composant. Ce nombre doit être compris entre 3 et 6.

Votre programme doit afficher le message `"Le nombre d'UE est défini"` ou, le cas échéant, l'un des messages suivants `"Le nombre d'UE est déjà défini"` ou `"Le nombre d'UE est incorrect"`.

Les commandes suivantes ne doivent être prises en compte par votre programme qu'une fois le nombre d'UE défini. Si ce n'est pas le cas, toutes les commandes suivantes doivent afficher le message `"Le nombre d'UE n'est pas défini"`.

### C3 - Ajout d'une épreuve

Une ligne composée de la chaîne de caractères `"épreuve"` suivie du numéro de semestre, du nom de la matière, du nom de l'épreuve, et d'un coefficient par UE. Le numéro de semestre doit être égal à 1 ou 2, chaque coefficient doit être supérieur ou égal à 0 et au moins un des coefficients doit être strictement supérieur à 0. De plus, aucune épreuve pré-existante dans ce semestre et cette matière ne doit porter le même nom.

Votre programme doit afficher le message `"Epreuve ajoutée à la formation"` ou, le cas échéant, l'un des messages suivants `"Le numéro de semestre est incorrect"`, `"Une même épreuve existe déjà"` ou `"Au moins un des coefficients est incorrect"`. De plus, en cas de succès et dans le cas où la commande correspond à la première épreuve de la matière, votre programme doit préalablement afficher le message `"Matière ajoutée à la formation"`.

## C4 - Vérification des coefficients

Une ligne composée de la chaîne de caractères `"coefficients"` suivie du numéro de semestre. Le numéro de semestre doit être égal à 1 ou 2.

Votre programme doit afficher le message `"Coefficients corrects"` ou, le cas échéant, l'un des messages suivants `"Le numéro de semestre est incorrect"`, `"Le semestre ne contient aucune épreuve"` ou `"Les coefficients d'au moins une UE de ce semestre sont tous nuls"`.

## C5 - Ajout d'une note

Une ligne composée de la chaîne de caractères `"note"` suivie du numéro de semestre, du nom d'un étudiant, du nom d'une matière, du nom d'une épreuve et d'une note. Le numéro de semestre doit être égal à 1 ou 2 et la note comprise entre 0 et 20.

Votre programme doit afficher le message `"Note ajoutée à l'étudiant"` ou, le cas échéant, l'un des messages suivants `"Le numéro de semestre est incorrect"`, `"Matière inconnue"`, `"Epreuve inconnue"`, `"Note incorrecte"` ou `"Une note est déjà définie pour cet étudiant"`. De plus, en cas de succès et dans le cas où la commande correspond à la première note de l'étudiant, votre programme doit préalablement afficher le message `"Etudiant ajouté à la formation"`.

## C6 - Vérification des notes

Une ligne composée de la chaîne de caractères `"notes"` suivie du numéro de semestre et du nom d'un étudiant. Le numéro de semestre doit être égal à 1 ou 2.

Votre programme doit afficher le message `"Notes correctes"` ou, le cas échéant, l'un des messages suivants `"Le numéro de semestre est incorrect"`, `"Etudiant inconnu"` ou `"Il manque au moins une note pour cet étudiant"`.

## C7 - Affichage d'un relevé de note

Une ligne composée de la chaîne de caractères `"relevé"` suivie du numéro de semestre et du nom d'un étudiant. Le numéro de semestre doit être égal à 1 ou 2.

Votre programme doit afficher le relevé de l'étudiant selon l'exemple donné ci-dessous ou, le cas échéant, l'un des messages suivants `"Le numéro de semestre est incorrect"`, `"Etudiant inconnu"`, `"Les coefficients de ce semestre sont incorrects"` ou `"Il manque au moins une note pour cet étudiant"`.

En admettant que les notes au semestre 1 de l'étudiant désigné par la commande sont celles présentées dans la Table 2, votre programme doit afficher :

	UE1	UE2	UE3
Programmation	10,0	10,2	ND
SGBD	13,0	ND	13,0
--			
Moyennes	11,2	10,2	13,0

Les alignements doivent être respectés et les notes doivent être arrondies sur une décimale (uniquement à l'affichage et pas lors des calculs). La note 9,99 doit être arrondie à 9,9 et non pas à 10.

## C8 - Affichage d'une décision de jury

Une ligne composée de la chaîne de caractères `"décision"` suivie du nom d'un étudiant.

Votre programme doit afficher les décisions de l'étudiant selon l'exemple donné ci-dessous ou, le cas échéant, l'un des messages suivants `"Le numéro de semestre est incorrect"`, `"Etudiant inconnu"`, `"Les coefficients d'au moins un semestre sont incorrects"` ou `"Il manque au moins une note pour cet étudiant"`.

En admettant que les relevés de l'étudiant désigné par la commande sont ceux présentés dans la Table 3, votre programme doit afficher :

	UE1	UE2	UE3
S1	11,2	10,2	13,0
S2	9,3	8,1	13,0
--			
Moyennes annuelles	10,2	9,1	13,0
Acquisition	UE1 et UE3		
Devenir	Passage		

Là encore, les alignements doivent être respectés et les mêmes arrondis réalisés. Les UE acquises doivent être séparées par une virgule suivie d'un espace à l'exception de la dernière UE qui doit être précédée de " et ". Dans le cas particulier où aucune UE est acquise, le message "Aucune" doit être affiché. Enfin, si la condition de passage n'est pas atteinte, le message "Redoublement" doit être affiché.

### 3 Spécifications techniques

Nous rappelons que toute formation est composée de 2 semestres et que le nombre d'UE est compris entre 3 et 6. De plus, le nombre de matières est limité à 10 par semestre. Chaque matière ne peut comporter plus de 5 épreuves. Pas plus de 100 étudiants suivent la formation. Enfin, la taille maximale des chaînes de caractères est de 30. Le dépassement de ces limites n'a pas à être testé.

Les messages attendus comprennent des lettres accentuées et le séparateur décimal des réels est la virgule. Pour ce faire, votre programme doit inclure le fichier d'entête `<locale.h>` et la première instruction de la fonction `main` doit être `setlocale(LC_ALL, "");`. Par exemple, le programme suivant

```
#include <locale.h> // pour setlocale
#include <stdio.h>
int main() {
    setlocale(LC_ALL, "");
    float pi = 3.1416;
    printf("Le nombre PI est évalué à %.2f", pi);
}
```

affiche "Le nombre PI est évalué à 3,14" (et non pas "Le nombre PI est évalué à 3.14").

### 4 Qui, quoi et quand?

Le projet est à réaliser en binôme (les monômes ne sont pas autorisés). Les membres d'un binôme seront de préférence du même groupe. Vous devez programmer et tester l'application demandée.

Le développement logiciel se fera par cycle de type agile au moyen de quatre sprints. Chaque sprint est défini par un ensemble de commandes devant être supporté par votre programme et un test par redirection correspondant à un jeu de données de test (JDT) (`in.txt`) et ses résultats attendus (`out.txt`). Les différents JDT seront déposés sur le disque COMMUN sous peu. Pour un sprint donné, si le résultat de votre application coïncide avec le résultat de référence (`out.txt`), le sprint est considéré comme étant validé. Vous passerez alors au développement du sprint suivant. Vous trouverez en annexe un exemple d'exécution du programme correspondant au dernier sprint. Les commandes à prendre en compte par les différents sprints sont les suivantes :

Sprint	Commandes
SP1	formation, épreuve, coefficient et exit
SP2	SP1 + note et notes
SP3	SP2 + relevé
SP4	SP3 + décision

Vous avez le droit de programmer d'autres commandes que celles demandées pour vous faciliter la mise au point du programme. Une commande permettant d'afficher les coefficients d'un semestre ou les notes d'un étudiant peut être judicieuse. Ces commandes supplémentaires ne seront pas testées lors de la recette (voir ci-dessous). Toutefois, il est fortement recommandé de mettre en avant ce travail dans votre rapport de développement (voir ci-dessous) si c'est le cas. Il en sera tenu compte lors de la notation.

Votre travail sera évalué à partir (1) d'une recette, (2) de la rédaction du dossier de développement logiciel et (3) d'une partie du DST de la ressource R1.01.

La semaine du **17 octobre 2022**, vous passerez la recette de votre développement logiciel lors de la seconde séance de TP. Il s'agit d'un test de recette qui testera le sprint de plus haut niveau atteint au cours de votre développement avec un nouveau JDT. Vous aurez à compiler et à exécuter votre programme avec le JDT de recette (type `in.txt`) qui vous sera communiqué. Votre enseignant vérifiera automatiquement la correction de votre application en comparant l'affichage produit au fichier de référence (`out.txt`). Si les 2 fichiers coïncident, le sprint est validé à la recette.

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est importante pour l'évaluation de votre travail. La composition de votre dossier doit être la suivante :

- Une page de garde indiquant le nom et le groupe des membres du binôme, l'objet du dossier. Une table des matières paginée sur l'ensemble du dossier incluant les annexes.
- Une brève présentation du projet (1 page) : le rôle fonctionnel de l'application, les entrées et sorties de l'application.

- L'organisation des tests de l'application et le bilan de validation des différents sprints que vous avez développés.
- Un bilan de projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).
- En annexe au dossier :
  1. Le listing complet de vos sources.
  2. la trace d'exécution du test du sprint de plus haut niveau atteint.

Les codes sources doivent être commentés (structures de données, champs, fonctions). Les conventions de nommage données en cours doivent être respectées. Les préconditions des fonctions doivent être documentées et testées par assertion dans le code. Suivez toutes les spécifications données et, en particulier, la composition du dossier de développement logiciel et la constitution de l'archive .zip demandée ci-dessous sous peine de pénalisation.

La date limite de remise de projet est fixée au **vendredi 21 octobre 2022**. Vous devrez déposer sur MOODLE une archive (.zip) constituée du dossier de développement (.pdf) et du source (uniquement le fichier C) du sprint de plus haut niveau que vous avez validé lors de la recette.

## Annexe

```

formation 3
Le nombre d'UE est défini
épreuve 1 Programmation Projet 1 2 0
Matière ajoutée
Epreuve ajoutée
épreuve 1 Programmation DST 2 3 0
Epreuve ajoutée
épreuve 1 SGBD Participation 0,5 0 0,5
Matière ajoutée
Epreuve ajoutée
épreuve 1 SGBD Rapport 1,5 0 1,5
Epreuve ajoutée
coefficients 1
Coefficients corrects
note 1 Paul Programmation Projet 12
Etudiant ajouté à la formation
Note ajoutée à l'étudiant
note 1 Paul Programmation DST 9
Note ajoutée à l'étudiant
note 1 Paul SGBD Participation 16
Note ajoutée à l'étudiant
note 1 Paul SGBD Rapport 12
Note ajoutée à l'étudiant
relevé 1 Paul

      UE1  UE2  UE3
Programmation 10,0 10,2  ND
SGBD          13,0  ND 13,0
--
Moyennes      11,2 10,2 13,0
épreuve 2 Architecture Interrogation 1 0 2
Matière ajoutée
Epreuve ajoutée

```

```

épreuve 2 Architecture DST 0 1 4
Epreuve ajoutée
épreuve 2 Système QCM 2 3 0,5
Matière ajoutée
Epreuve ajoutée
épreuve 2 Système Exposé 3 2 0,5
Epreuve ajoutée
coefficients 2
Coefficients corrects
note 2 Paul Architecture Interrogation 18
Note ajoutée à l'étudiant
note 2 Paul Architecture DST 12
Note ajoutée à l'étudiant
note 2 Paul Système QCM 7
Note ajoutée à l'étudiant
note 2 Paul Système Exposé 8
Note ajoutée à l'étudiant
relevé 2 Paul

      UE1  UE2  UE3
Architecture 18,0 12,0 14,0
Système      7,6  7,4  7,5
--
Moyennes      9,3  8,1 13,0
décision Paul

      UE1  UE2  UE3
S1          11,2 10,2 13,0
S2           9,3  8,1 13,0
--
Moyennes annuelles 10,2  9,1 13,0
Acquisition        UE1 et UE3
Devenir            Passage
exit

```

Figure 1: Exemple de session – en rouge, les données saisies par l'utilisateur, en bleu, les messages affichés par le programme.