

BUT1 – SAÈ S1.01

IMPLÉMENTATION D'UN BESOIN CLIENT

– GESTION DE LA SCOLARITÉ DES ÉTUDIANTS DE BUT –

L'objectif du projet est de développer un interpréteur de commande gérant les jurys d'une formation BUT Informatique.

1 Règles de gestion

Pour simplifier, nous ferons l'hypothèse que les semestres sont tous composés de 6 unités d'enseignement (alors que dans la réalité ceux de la 3ème année n'en comportent que 3). De même, nous ne prendrons pas en compte les possibilités de passage par décision de jury, les redoublements ainsi que les passerelles entrantes et sortantes (i.e. arrivées et départs d'étudiants en cours de formation).

Chaque semestre, les 6 unités d'enseignement (UE) sont évaluées par une note comprise entre 0 et 20. A la fin de l'année, les notes du semestre impair et du semestre pair de chaque UE sont moyennées. Cette moyenne est dite être la note du regroupement cohérent d'unités d'enseignement (RCUE).

Règles de validation des UE :

- Les UE pour lesquelles l'étudiant a obtenu une note supérieure ou égale à 10 est validée.
- De même, les UE appartenant à un RCUE auquel l'étudiant a obtenu une note supérieure ou égale à 10 sont elles aussi validées (règle de compensation).
- Une note de RCUE supérieure ou égale à 10 valide automatiquement les UE correspondantes de l'année précédente.

Règles de passage et d'obtention du diplôme

- A tout moment de leur scolarité, un étudiant peut démissionner ou être déclaré défaillant (en cas d'absences non justifiées répétées). Les démissionnaires et défaillants quittent la formation.
- En fin de semestre impair, les étudiants passent de droit dans le semestre suivant et cela quelque soit leurs résultats.
- En fin de semestre pair, ils doivent avoir obtenu une note supérieure à 10 à plus de la moitié des RCUE et aucune strictement inférieure à 8. Dans le cas contraire, les étudiants sont ajournés et quittent la formation.
- Pour pouvoir passer en 3ème année, toutes les UE de la première année doivent avoir été validées. Les étudiants ne satisfaisant pas cette condition sont ajournés et quittent la formation.
- Seuls les étudiants ayant validé toutes les UE des 3 années se voient délivrer le diplôme de fin de formation. Les autres sont ajournés et quittent 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'. Toute commande se termine par un retour à la ligne.

Votre programme doit lire les commandes sur l'entrée standard (le clavier) et produire des résultats sur la sortie standard (l'écran). Les commandes saisies sont supposées respecter le format attendu. Seules les erreurs explicitement indiquées dans ce document doivent être prises en compte par votre programme.

Les commandes sont les suivantes.

EXIT - Sortie du programme

Une ligne composée de la chaîne de caractères "EXIT". Elle doit provoquer la fin de l'exécution du programme.

INSCRIRE - Inscription d'un étudiant

Une ligne composée de la chaîne de caractères "INSCRIRE" suivie du prénom et du nom de l'étudiant. Le prénom et le nom sont des chaînes de caractères ne dépassant pas 30 caractères et ne contenant pas d'espace.

Votre programme doit afficher le message "Inscription enregistree" suivi d'un identifiant entre parenthèses. Celui-ci doit correspondre au nombre d'étudiants connus (1 pour le premier étudiant, 2 pour le suivant, ...).

Si un étudiant déjà connu porte le même nom et le même prénom, votre programme doit afficher le message "Nom incorrect".

Dans tous les exemples qui suivent, le texte en **rouge** correspond aux données saisies par l'utilisateur alors que le texte en **bleu** correspond aux messages affichés par le programme.

```
INSCRIRE Marcel Pagnol
Inscription enregistrée (1)
```

Le message affiché nous indique qu'aucun étudiant n'était connu jusqu'alors et qu'en conséquence, il sera désigné par la suite avec l'identifiant 1. Il est bien le premier.

CURSUS - Cours d'un étudiant

Une ligne composée de la chaîne de caractères "CURSUS" suivie de l'identifiant de l'étudiant. L'identifiant est un entier devant désigner un étudiant connu.

Votre programme doit afficher le nom et le prénom de l'étudiant sur une première ligne. De plus, une ligne supplémentaire par semestre/année (passé et courant) doit être affichée. Cette ligne doit préciser le code du semestre (S1, S2, ...) ou de l'année (B1, B2, ...) ainsi que les notes d'UE ou de RCUE.

Le cas échéant, votre programme doit afficher le message "Identifiant incorrect" si celui-ci ne désigne pas un étudiant connu.

```
CURSUS 1
Marcel Pagnol
S1 - * (*) - * (*) - * (*) - * (*) - * (*) - * (*) - en cours
```

On remarque que l'étudiant nouvellement inscrit a été automatiquement affecté au semestre 1 (S1). Pour l'instant, les notes d'UE obtenues par cet étudiant à ce semestre sont inconnues. Cela est représenté par le caractère "*". Nous verrons plus loin que chaque note est suivie d'un code mis entre parenthèses. Pour les notes inconnues, le code est là encore une étoile ("*"). La chaîne "en cours" indique que le semestre 1 de cet étudiant n'est pas encore terminé.

Nous verrons le format précis des messages affichés par cette commande au travers des différents exemples donnés ci-dessous.

NOTE - Note d'une UE

Une ligne composée de la chaîne de caractères "NOTE" suivie de l'identifiant de l'étudiant, du numéro d'UE et de la note. Le numéro d'UE est un entier compris entre 1 et 6. La note est un réel compris entre 0 et 20.

Votre programme doit afficher le message "Note enregistrée". Si l'étudiant est inconnu, votre programme doit afficher le message "Identifiant incorrect" et, si le dernier semestre suivi par l'étudiant n'est pas "en cours", le message "Etudiant hors formation".

```
NOTE 1 2 10.5
Note enregistrée
NOTE 1 4 8.5
Note enregistrée
NOTE 1 6 16.5
Note enregistrée
NOTE 1 6 6.5
Note enregistrée
CURSUS 1
Marcel Pagnol
S1 - * (*) - 10.5 (ADM) - * (*) - 8.5 (AJ) - * (*) - 6.5 (AJ) - en cours
```

L'étudiant portant le numéro 1 a obtenu la note de 10,5 à l'UE 2 et cette note a bien été enregistrée. Il en est de même pour son 6,5 à l'UE 4 et son 6,5 à l'UE 6. L'utilisateur a fait une faute de frappe en saisissant cette dernière note en lui affectant 16,5. Il l'a corrigé avec une commande NOTE supplémentaire. Les notes connues du semestre en cours peuvent être modifiées.

La commande CURSUS nous permet de constater que ces 3 notes ont bien été enregistrées. Les notes supérieures ou égales à 10 ont le code ADM, celles strictement plus petites que 10 ont le code AJ. On remarque que les notes des UE 1, 3 et 5 sont toujours inconnues.

DEMISSION/DEFAILLANCE - Démission et défaillance d'un étudiant

Une ligne composée de la chaîne de caractères "DEMISSION" ou "DEFAILLANCE" suivie de l'identifiant d'un étudiant.

Le cas échéant, votre programme doit afficher le message "Identifiant incorrect" si l'étudiant est inconnu ou "Etudiant hors formation" si le dernier semestre suivi par l'étudiant n'est pas "en cours". Sinon, votre programme doit afficher le message "Démission enregistrée" ou "Défaillance enregistrée".

Un étudiant est noté défaillant si son nombre d'absences injustifiées est trop important. Dans ce cas, et comme pour un démissionnaire, la scolarité de l'étudiant est stoppée, il est considéré hors formation et ses notes ne peuvent plus être modifiées dans l'application.

DEMISSION 1

Demission enregistree

CURSUS 1

Marcel Pagnol

S1 - * (*) - 10.5 (ADM) - * (*) - 8.5 (AJ) - * (*) - 6.5 (AJ) - demission

Le mot "demission" affiché par la commande CURSUS indique que Marcel a démissionné. Le mot "defaillance" est employé pour les étudiants défaillants. L'étudiant reste connu de l'application mais sa situation ne peut plus changer. En particulier, les commandes NOTE, DEMISSION et DEFAILLANCE concernant cet étudiant doivent afficher "Etudiant hors formation". Nous verrons plus loin qu'il en sera de même avec les étudiants ajournés ou déjà diplômés.

JURY - Jury de fin de semestre

Une ligne composée de la chaîne de caractères "JURY" suivie d'un numéro de semestre. Ce numéro est un entier compris entre 1 et 6.

Le cas échéant, votre programme doit afficher le message "Semestre incorrect". De même, il doit afficher "Des notes sont manquantes" si au moins un des étudiants "en cours" de ce semestre a encore une note inconnue à une UE. Dans le cas contraire, il doit afficher "Semestre termine pour X etudiant(s)" où X est remplacé par le nombre d'étudiants pour lesquels une décision de jury a été prise.

...

CURSUS 2

2 Kilian Mbappe

S1 - 10.5 (ADM) - 15.2 (ADM) - 10.1 (ADM) - 11.7 (ADM) - 5.0 (AJ) - 17.7 (ADM) - en cours

CURSUS 3

3 Ousmane Dembele

S1 - 13.0 (ADM) - 11.2 (ADM) - 10.0 (ADM) - 12.8 (ADM) - 14.5 (ADM) - 12.1 (ADM) - en cours

Les commandes ayant conduit à la situation ci-dessus ont été omises. Nous faisons l'hypothèse que Kilian et Ousmane sont les 2 seuls étudiants en cours du semestre 1.

En fin de semestre impair tous les étudiants passent dans le semestre suivant.

JURY 1

Semestre termine pour 2 etudiant(s)

CURSUS 2

2 Kilian Mbappe

S1 - 10.5 (ADM) - 15.2 (ADM) - 10.1 (ADM) - 11.7 (ADM) - 5.0 (AJ) - 17.7 (ADM) -

S2 - * (*) - * (*) - * (*) - * (*) - * (*) - * (*) - en cours

CURSUS 3

3 Ousmane Dembele

S1 - 13.0 (ADM) - 11.2 (ADM) - 10.0 (ADM) - 12.8 (ADM) - 14.5 (ADM) - 12.1 (ADM) -

S2 - * (*) - * (*) - * (*) - * (*) - * (*) - * (*) - en cours

Nous pouvons remarquer qu'à présent les 2 étudiants sont bien inscrits au semestre 2 et que les notes d'UE de ce semestre attendent d'être saisies.

NOTE 2 3 8.3

Note enregistree

CURSUS 2

2 Kilian Mbappe

S1 - 10.5 (ADM) - 15.2 (ADM) - 10.1 (ADM) - 11.7 (ADM) - 5.0 (AJ) - 17.7 (ADM) -

S2 - * (*) - * (*) - 8.3 (AJ) - * (*) - * (*) - * (*) - en cours

Les nouvelles notes sont bien affectées au semestre en cours de l'étudiant.

...

CURSUS 2

2 Kilian Mbappe

S1 - 10.5 (ADM) - 15.2 (ADM) - 10.1 (ADM) - 11.7 (ADM) - 5.0 (AJ) - 17.7 (ADM) -

S2 - 9.7 (AJ) - 13.6 (ADM) - 8.3 (AJ) - 12.7 (ADM) - 15.2 (ADM) - 15.5 (ADM) - en cours

CURSUS 3

3 Ousmane Dembele

S1 - 13.0 (ADM) - 11.2 (ADM) - 10.0 (ADM) - 12.8 (ADM) - 14.5 (ADM) - 12.1 (ADM) -

S2 - 14.0 (ADM) - 9.6 (AJ) - 5.8 (AJ) - 11.8 (ADM) - 12.7 (ADM) - 18.7 (ADM) - en cours

À présent, nos deux étudiants ont des notes à chacune des UE du deuxième semestre. Le jury de **fin de semestre pair** peut avoir lieu.

JURY 2

Semestre termine pour 2 etudiant(s)

CURSUS 2

2 Kilian Mbappe

S1 - 10.5 (ADM) - 15.2 (ADM) - 10.1 (ADM) - 11.7 (ADM) - 5.0 (ADC) - 17.7 (ADM) -

S2 - 9.7 (ADC) - 13.6 (ADM) - 8.3 (AJ) - 12.7 (ADM) - 15.2 (ADM) - 15.5 (ADM) -

B1 - 10.1 (ADM) - 14.4 (ADM) - 9.2 (AJ) - 12.2 (ADM) - 10.1 (ADM) - 16.6 (ADM) -

S3 - * (*) - * (*) - * (*) - * (*) - * (*) - * (*) - en cours

CURSUS 3

3 Ousmane Dembele

S1 - 13.0 (ADM) - 11.2 (ADM) - 10.0 (ADM) - 12.8 (ADM) - 14.5 (ADM) - 12.1 (ADM) -

S2 - 14.0 (ADM) - 9.6 (ADC) - 5.8 (AJ) - 11.8 (ADM) - 12.7 (ADM) - 18.7 (ADM) -

B1 - 13.5 (ADM) - 10.6 (ADM) - 7.9 (AJB) - 12.3 (ADM) - 13.6 (ADM) - 15.4 (ADM) - ajourne

Kilian passe alors que Ousmane est ajourné. La commande **CURSUS** explique la situation de chacun. La ligne signalée par B1 représente le bilan de la première année. Pour chaque UE est calculée la moyenne des notes obtenues à chaque semestre de l'année, *i.e.* la note de RCUE.

Par exemple, Kilian a eu 10,5 à l'UE 1 au semestre 1 et 9,7 à cette même UE au semestre suivant. Le RCUE correspondant est donc 10,1 $((10,5 + 9,7)/2)$. Cette note est supérieure ou égale à 10 et est donc qualifiée du code ADM. Les notes comprises entre 8 (inclus) et 10 (exclu) portent le code AJ et celles strictement inférieures à 8 le code AJB pour signifier que la borne des 8 n'a pas été atteinte.

De plus, certains codes associés aux UE des semestres de l'année ont été modifiés. Les codes AJ des UE dont la note de RCUE est supérieure à 10 ont été transformés en ADC pour signifier que l'UE est validée par compensation même si la note est inférieure à 10.

Pour Kilian, c'est le cas pour l'UE 5 du semestre 1 et pour l'UE 1 du semestre 2. Par contre, la note d'UE 3 du semestre 2 est toujours AJ car la note du RCUE correspondant (9,2) n'est pas suffisante.

Un étudiant passe dans l'année supérieure s'il a validé (code ADM) plus de la moitié des RCUE (et donc au moins 4) et qu'aucun RCUE n'a une note strictement inférieure à 8 (code AJB). Kilian passe avec une dette (l'UE 3 qu'il devra valider lors de la prochaine année) alors que Ousmane est ajourné pour un dixième de point à l'UE 3. Sa scolarité est terminée et ses notes ne peuvent plus être modifiées.

De nouvelles règles s'appliquent pour le jury de **fin de deuxième année**.

...

CURSUS 2

2 Kilian Mbappe

S1 - 10.5 (ADM) - 15.2 (ADM) - 10.1 (ADM) - 11.7 (ADM) - 5.0 (ADC) - 17.7 (ADM) -

S2 - 9.7 (ADC) - 13.6 (ADM) - 8.3 (AJ) - 12.7 (ADM) - 15.2 (ADM) - 15.5 (ADM) -

B1 - 10.1 (ADM) - 14.4 (ADM) - 9.2 (AJ) - 12.2 (ADM) - 10.1 (ADM) - 16.6 (ADM) -

S3 - 10.0 (ADM) - 12.0 (ADM) - 9.5 (AJ) - 10.0 (ADM) - 13.0 (ADM) - 15.0 (ADM) -

S4 - 9.2 (AJ) - 13.0 (ADM) - 11.1 (ADM) - 9.0 (AJ) - 12.0 (ADM) - 16.0 (ADM) - en cours

Les notes obtenues par Kilian au quatrième semestre sont complètes et le jury peut se tenir. De manière similaire à la première année, des UE des semestres 3 et 4 initialement codées AJ peuvent être transformées en ADC par application de la règle de compensation. De plus, les résultats de la première année peuvent eux aussi être modifiés. La validation d'un RCUE de la deuxième année conduit automatiquement à la validation du RCUE correspondant de l'année précédente (ainsi que des UE associées). Ainsi, les UE et RCUE portant le code AJ de l'année précédente sont transformés en ADS pour signifier leur validation par les résultats de l'année suivante.

Enfin, un étudiant ne peut passer en troisième année que si toutes les UE de la première année sont validées (codes ADM, ADC ou ADS).

JURY 4

Semestre termine pour 1 etudiant(s)

CURSUS 2

2 Kilian Mbappe

S1 - 10.5 (ADM) - 15.2 (ADM) - 10.1 (ADM) - 11.7 (ADM) - 5.0 (ADC) - 17.7 (ADM) -
 S2 - 9.7 (ADC) - 13.6 (ADM) - 8.3 (ADS) - 12.7 (ADM) - 15.2 (ADM) - 15.5 (ADM) -
 B1 - 10.1 (ADM) - 14.4 (ADM) - 9.2 (ADS) - 12.2 (ADM) - 10.1 (ADM) - 16.6 (ADM) -
 S3 - 10.0 (ADM) - 12.0 (ADM) - 9.5 (ADC) - 10.0 (ADM) - 13.0 (ADM) - 15.0 (ADM) -
 S4 - 9.2 (AJ) - 13.0 (ADM) - 11.1 (ADM) - 9.0 (AJ) - 12.0 (ADM) - 16.0 (ADM) -
 B2 - 9.6 (AJ) - 12.5 (ADM) - 10.3 (ADM) - 9.5 (AJ) - 12.5 (ADM) - 15.5 (ADM) -
 S5 - * (*) - * (*) - * (*) - * (*) - * (*) - * (*) - en cours

Sa note à la RCUE 3 (10, 3) lui permet de valider l'UE 3 du semestre 3 (note 9, 5), le code AJ est transformé en ADC. De plus, cela provoque le remboursement de sa dette de sa première année à cette compétence. Les codes AJ de l'UE 3 du deuxième semestre et du RCUE 3 de la première année ont été transformés en ADS.

Kilian valide 4 RCUE sur 6 (code ADM) ainsi que toutes les UE de la première année (codes ADM, ADC ou ADS). En conséquence, il est admis en troisième année. Il est automatiquement inscrit au cinquième semestre et l'application est prête à enregistrer les notes de ce semestre.

Lors du jury de **fin de scolarité** (jury du semestre 6), les étudiants ayant validé toutes les UE des trois années sont diplômés. Les autres sont ajournés. Dans tous les cas, les résultats de ces étudiants ne peuvent plus être modifiés.

Regardons les résultats de notre étudiant fétiche.

JURY 6

Semestre termine pour 1 etudiant(s)

CURSUS 2

2 Kilian Mbappe

S1 - 10.5 (ADM) - 15.2 (ADM) - 10.1 (ADM) - 11.7 (ADM) - 5.0 (ADC) - 17.7 (ADM) -
 S2 - 9.7 (ADC) - 13.6 (ADM) - 8.3 (ADS) - 12.7 (ADM) - 15.2 (ADM) - 15.5 (ADM) -
 B1 - 10.1 (ADM) - 14.4 (ADM) - 9.2 (ADS) - 12.2 (ADM) - 10.1 (ADM) - 16.6 (ADM) -
 S3 - 10.0 (ADM) - 12.0 (ADM) - 9.5 (ADC) - 10.0 (ADM) - 13.0 (ADM) - 15.0 (ADM) -
 S4 - 9.2 (ADS) - 13.0 (ADM) - 11.1 (ADM) - 9.0 (ADS) - 12.0 (ADM) - 16.0 (ADM) -
 B2 - 9.6 (ADS) - 12.5 (ADM) - 10.3 (ADM) - 9.5 (ADS) - 12.5 (ADM) - 15.5 (ADM) -
 S5 - 1.0 (ADC) - 12.0 (ADM) - 11.0 (ADM) - 12.0 (ADM) - 13.0 (ADM) - 16.0 (ADM) -
 S6 - 19.0 (ADM) - 14.0 (ADM) - 14.0 (ADM) - 11.0 (ADM) - 11.0 (ADM) - 14.0 (ADM) -
 B3 - 10.0 (ADM) - 13.0 (ADM) - 12.5 (ADM) - 11.5 (ADM) - 12.0 (ADM) - 15.0 (ADM) - diplome

Kilian a bien remboursé les deux dettes de l'année précédente (RCUE 1 et 4), il a bien compensé son accident à l'UE 1 du cinquième semestre. Toutes les UE de tous les semestres sont validés (codes ADM, ADC ou ADS). Il est donc diplômé. Dans le cas contraire, la ligne étiquetée B3 doit se terminer par le mot "ajourne".

ETUDIANTS - Liste des étudiants

Une ligne composée de la chaîne de caractères "ETUDIANTS".

Votre programme doit afficher la liste des étudiants connus. Pour chacun doit figurer son identifiant, son prénom et son nom, le dernier semestre atteint et son statut (en cours, démission, défaillance, diplomation ou ajournement). Le format de l'affichage est donné dans l'exemple ci-dessous. Les étudiants sont triés par identifiant croissant.

ETUDIANTS

1 - Marcel Pagnol - S1 - demission
 2 - Kilian Mbape - S6 - diplome
 3 - Ousmane Dembele - S2 - ajourne

BILAN - Bilan d'une année

Une ligne composée de la chaîne de caractères "BILAN" suivie d'un numéro d'année. Ce numéro est compris entre 1 et 3.

Le cas échéant, votre programme doit afficher le message "Annee incorrecte" si le numéro ne désigne pas une année correcte. Dans le cas contraire, le programme affiche le décompte de tous les étudiants ayant atteint un semestre de cette année en distinguant leur statut :

- Le nombre de démissionnaires (à l'un ou l'autre des semestres de l'année)
- Le nombre de défaillants (à l'un ou l'autre des semestres de l'année)
- Le nombre d'étudiants pour lesquels le jury de fin de cette année n'a pas encore eu lieu (en cours)
- Le nombre d'étudiants ajournés
- Le nombre d'étudiants qui sont passés dans l'année supérieure (ou ont été diplômés pour la dernière année)

Le format d'affichage est donné dans l'exemple qui suit.

```

BILAN 1
1 demission(s)
0 defaillance(s)
0 en cours
1 ajourne(s)
1 passe(s)

```

3 Limites numériques, les notes, une précision, un jeu d'essai et un conseil

3.1 Limites numériques

Il y a 6 UE et 3 années (et donc 6 semestres). Le nombre d'étudiants est limité à 100. Les prénoms et noms des étudiants ne dépassent pas 30 caractères.

3.2 Les notes

Les notes saisies sont des réels compris entre 0 et 20. Par exemple, 10, 10,5 et 10,58 sont des notes légales et doivent être mémorisées par votre application sous la forme de `float`. Le nombre de décimales ne doit pas être limité et les seules erreurs d'arrondi dans le calcul des moyennes doivent provenir des limites du type.

Par contre, les notes doivent être affichées avec une seule décimale et sans faire d'arrondi. Les décimales en trop doivent être tronquées. La fonction `floor` de la bibliothèque standard (inclure `math.h`) permet d'obtenir le bon format. Par exemple, `float note = 10.58; printf("%.1f", floor(note * 10.) / 10.);` affiche 10.5.

3.3 Une précision

Les commandes décrites ci-dessus permettent des usages qui peuvent paraître étonnants. Par exemple, des étudiants en cours de deuxième semestre peuvent être rejoints par de nouveaux étudiants de premier semestre suite à un jury de celui-ci.

```

...
ETUDIANTS
1 - Mikal Ziane - S1 - en cours
2 - Denis Poitreaud - S2 - en cours
CURSUS 1
1 Mikal Ziane
S1 - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - en cours
CURSUS 2
2 Denis Poitreaud
S1 - 20.0 (ADM) - 20.0 (ADM) - 20.0 (ADM) - 20.0 (ADM) - 20.0 (ADM) - 20.0 (ADM) -
S2 - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - en cours

```

Les jurys des deux semestres peuvent se tenir, tous les étudiants ont des notes à chacune des UE de leur semestre en cours.

```

JURY 1
Semestre termine pour 1 etudiant(s)
CURSUS 1
1 Mikal Ziane
S1 - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) - 0.0 (AJ) -
S2 - * (*) - * (*) - * (*) - * (*) - * (*) - * (*) - en cours

```

Le jury du semestre 2 ne peut plus se tenir à présent car au moins une note d'UE d'au moins un étudiant (Mikal) n'est pas connue. Ce comportement est normal et doit être supporté par votre application.

3.4 Un jeu d'essai

L'ensemble des commandes à saisir pour reproduire l'exemple fil rouge du sujet est fourni au sein du fichier `in.txt`. Vous pouvez vous en servir pour tester votre programme. La suite des messages que devrait afficher votre programme en réaction à ces commandes est réunie au sein du fichier `out.txt`. Vous pouvez vous servir de celui-ci pour vérifier que votre application fonctionne comme attendu.

Pour vous permettre de repérer facilement à quelle commande correspondent des messages, le fichier `inout.txt` mixte la séquence des 2 fichiers ci-dessus¹. Les commandes y sont précédées du caractère `>` pour les distinguer des messages affichés.

3.5 Un conseil

Dans votre application, il est recommandé de ne pas mémoriser des données si celles-ci peuvent être calculées à partir d'autres informations déjà mémorisées.

4 Qui, quoi et quand?

Le projet est à réaliser en binôme (les monômes peuvent être autorisés exceptionnellement si vous le demandez). 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.

Les commandes à prendre en compte par les différents sprints sont les suivantes :

Sprint	Commandes
SP1	EXIT, INSCRIRE, NOTE, ETUDIANTS
SP2	SP1 + DEMISSION, DEFAILLANCE, JURY semestre impair
SP3	SP2 + JURY semestre pair
SP4	SP3 + BILAN

Vous avez le droit de programmer d'autres commandes que celles demandées pour vous faciliter la mise au point du programme. 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 rapport et (3) d'une partie du DST de la ressource R1.01.

La semaine du **20 octobre 2025**, 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 deux coïncident, le sprint sera considéré comme étant validé pour la recette.

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

- Une page de garde indiquant le nom et le groupe des membres du binôme, l'objet du rapport. Une table des matières paginée sur l'ensemble du dossier incluant les annexes.
- Une brève présentation du projet (1 page maximum) : le rôle fonctionnel de l'application, les entrées et sorties de l'application.
- Une présentation succincte des moyens que vous aurez mis en œuvre pour vous assurer de la correction de votre programme.
- Un bilan du projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).
- En annexe au dossier :
 1. Le code source (commenté et bien indenté) de votre programme.
 2. une courte 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 24 octobre 2025**. 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.

¹Les fichiers `in.txt`, `out.txt` et `inout.txt` correspondant sont joints au présent document et sont accessibles dans les signets. Toutefois, tous les *viewers PDF* ne supportent pas l'extraction de fichiers attachés. Acroread et Firefox le supportent.