

 Université Paris Cité	BUT 1 Groupe 110 XU Sarah Groupe 108 XU Célia
--	--

BUT1 – SAE S1.01

Implémentation d'un besoin client

– Gestion des absences –

Rapport du projet C

Le présent dossier, élaboré dans le cadre de notre projet de développement informatique, fournit une analyse complète et structurée de notre travail. Ce document commence par une page de garde détaillant les noms des membres du binôme et l'objet du dossier, suivie d'une table des matières. Une introduction présente le projet. Une section dédiée expose nos tests unitaires, mettant en lumière ceux réussis et ceux échoués. Le bilan du projet aborde les défis rencontrés, nos succès, et les perspectives d'amélioration. Enfin, une annexe ordonnée contient l'intégralité du code source et la trace du sprint le plus élevé.

TABLE DES MATIERES

I – PRESENTATION DU PROJET	3
a – Rôle fonctionnel du projet.....	3
b – Entrées et sorties de l’application	3
II – ELEMENTS DE CORRECTION	4
a – Fichier in/out.....	4
b – Travail de groupe	5
III – BILAN DU PROJET.....	6
a – Difficultés rencontrées.....	6
b – Réussites.....	10
c – Pistes d’améliorations.....	11
ANNEXES	12
Code Source de notre Programme	12
Trace d’exécution du test du sprint (3) de plus haut niveau atteint :	24

I – Présentation du Projet

a – Rôle fonctionnel du projet

Le programme effectué au cours de ce projet de Saé est un interpréteur de commandes : ici 9 commandes distinctes entrées au clavier par l'utilisateur. Il a été réalisé en C au cours de la période A du premier semestre 2024. Ce dernier permet de gérer les absences des étudiants d'une formation BUT et de les enregistrer. En effet, l'utilisateur peut enregistrer de nouveaux étudiants par le biais de la commande C1 pour les inscrire. Ensuite, ces étudiants peuvent avoir des absences qu'il faudra enregistrer. Cette fonctionnalité est prise en charge par la commande C2. Une fonction étudiante C3 permet d'afficher l'ensemble des étudiants étant inscrits dans la formation BUT en précisant leur nombre d'absences. Une fonction C4 permet d'enregistrer le dépôt d'un justificatif d'une absence. La fonction validations C5 permet au directeur de voir les justificatifs des étudiants en attente de validation. La commande C6 permet au directeur d'études de valider ou invalider un justificatif déposé dans les temps¹. La fonction C7 permet de classer les absences d'un étudiant par état ("En attente justificatif", "En attente validation", "Justifiées", "Non-justifiées") et trie chaque section par ordre chronologique. Enfin, seules les sections contenant des absences avant le jour actuel sont affichées, avec le détail de chaque absence. La fonction défaillante C8 permet d'afficher une liste de l'ensemble des étudiants avec leur nombre total d'absences ayant au moins 5 absences non-justifiées avant la date courante.

b – Entrées et sorties de l'application

Commande	Entrées	Sortie
C1	« Inscription » + « Nom » + NumGroupe	Inscription enregistrée (id)
C2	« absence » + id + NumJour + « demiJour »	« Absence enregistrée [idAbs] »
C3	« étudiants » + JourCourant	Liste détails étudiants avec total d'absences
C4	« justificatif » + idAbs + « JourJusti » + « explication »	« Justificatif enregistré »
C5	« validations »	Liste des absences en attente de validations
C6	« validation » + l'idAbs + « JustiAcceptée »	« Validation enregistrée »
C7	« étudiant » + id + JourCourant	Détails de l'étudiant (selon le format de la commande C3) et la liste des absences de l'étudiant selon leur état
C8	« défaillants » + JourCourant	Liste des étudiants défaillants (selon le format de la commande C3)
C0	« exit »	Quitte le programme

¹ Dispose de 3 jours après l'absence pour justifier

II – Éléments de correction

a – Fichier in/out

Nous avons corrigé notre code au fur et à mesure des fonctions. Nous avons mis en place plusieurs tests à la main pour vérifier précisément la cohérence de la réponse qui devrait être sortie.

Exemple de création de tests :

IN

Par exemple :

inscription Lea 106

inscription JP 102

absence 1 1 am

justificatif 1 3 je me suis endormi dans le métro

etudiant 1 1

etudiant 1 3

validation 1 ko

etudiant 1 1

etudiant 1 3

OUT

Inscription enregistree (1)

Inscription enregistree (2)

Absence enregistree [1]

Justificatif enregistre

(1) Lea 106 1

- En attente justificatif

[1] 1/am

XU-108 XU-110

(1) Lea 106 1

- En attente validation

[1] 1/am (je me suis endormi dans le métro)

Validation enregistrée

(1) Lea 106 1

- En attente justificatif

[1] 1/am

(1) Lea 106 1

- Non-justifiées

[1] 1/am (je me suis endormi dans le métro)

Dès que nous pensions qu'une fonction était finalisée, nous avons procédé aux différents sprints donnés par le professeur : d'abord le in du sujet puis le sprint normal ensuite le sprint sans erreur et enfin avec erreur. Pour vérifier les in et out facilement, nous avons utilisés les commandes de l'invite de commandes afin de créer un fichier sorti et ensuite de le comparer avec le sprint out donné, car il était très difficile de les comparer à l'aide des yeux en raison de la longueur trop importante des sprints. Nous avons utilisé : `fc /w res-xxx.txt out-xxx.txt`. Et dès qu'une fonction à passer tous les sprints, nous avons fait les fonctions suivantes.

Mais lorsqu'une différence a été trouvée, nous avons cherché à la main, chaque OUT reliait à un IN pour voir pourquoi notre commande ne sortait pas la même chose et en fonction de cela, nous avons corrigé les incohérences. Nous avons également fait des `printf` à des endroits précis pour vérifier que l'entrée de l'utilisateur rentre bien dans les boucles ou dans les `if` où ils sont censés entrer.

b – Travail de groupe

Une consigne qui a été mal interprétée était celle de la commande « validation » (C6) : si l'identifiant ne désigne pas une absence connue OU si l'absence n'est pas en attente de validation" alors le message "Identifiant incorrect" doit être affiché. Grâce aux questions de nos camarades et à la réponse du professeur sur Discord, nous avons pu mieux comprendre la consigne et ainsi corriger très vite le problème d'affichage d'erreur.

Nous avons observé des problèmes. Pour mieux les localiser nous avons utilisés le débogueur pour trouver l'endroit exact où une erreur se produisait.

De plus, les deux dernières semaines avant la formation de la recette de notre programme, nous avons corrigé une erreur de redondance. Nous avons fait au départ 4 structures : Etudiant, Absence, Justificatif et Validation. Mais nous nous sommes rendu compte qu'il n'était pas très utile de créer autant de structures, car en plus de cela, nous devons joindre chaque structure de

table, et cela engendrait beaucoup plus de ligne de code à programmer. Pour éviter les erreurs inutiles, nous avons réduit le nombre de structures en deux structures : Etudiant et Absence. Ce qui est beaucoup plus cohérent.

Malgré nos vérifications avec les sprints et le débogueur, des problèmes subsistaient, lorsqu'une incohérence ou qu'un bug se présentait nous nous envoyions notre version du code afin que l'autre personne puisse chercher et aider à trouver une solution.

Nous avons aussi pu poser des questions à nos professeurs lors des Travaux Pratiques (TP).

III – Bilan du projet

a – Difficultés rencontrées

Notre première difficulté eut été de trouver une structure idéale afin de stocker toutes les données du sujet, en effet nous avons dû modifier ces structures de multitudes de fois au fil du sujet et nous avons sûrement créés des variables superflues qui pourraient donc optimiser le programme.

La plupart de nos erreurs se trouvaient dans « etudiant », cette fonction est très utilisée et est donc une source d'erreur plus probable. Cette commande a en effet des répercussions directes dans les fonctions « justificatif », « validations », « validation » et « defaillants », car selon l'ordre d'exécutions des entrées, les états des absences changent à chaque appel, rendant des absences déjà justifiés en attente de justificatifs par exemple. Nous avons rencontré des problèmes au niveau de l'affichage des états des absences, ces derniers devaient être changés en fonction de plusieurs facteurs que nous devons réfléchir très minutieusement pour ne pas en délaissier une seule qui mène au changement de l'état. Nous avons énuméré au brouillon tous les cas possibilités pour qu'une absence change d'état. Mais malgré cela, nous avons réussi tous les sprints sauf le sprint 4 complémentaire avec erreur. Après plusieurs jours de réflexion, dans la fonction « etudiant », nous avons eu l'idée de faire une copie du tabAbs dans un deuxième tableau tab2 pour pouvoir changer les états dans le tab2 et laisser les états du tabAbs. Pour que le problème de « voyage dans le temps » soit résolu. Et faire attention au dépassement de la pile en initialisant le tab2.

Code changé de la fonction C7 étudiant (en factorisant en sous-fonction et en réglant le problème de ne pas changer l'état général des étudiants :

```
enum { MAX_ETAT = 4 };
/*XXXXXXXXXXXXXXXXXXXXX Sous Fonction Compare 3 XXXXXXXXXXXXXXXXXXXXXXXX*/
int compare3(const void* a, const void* b) {
    //conversion
    const Absence* abs1 = (const Absence*)a;
    const Absence* abs2 = (const Absence*)b;
    if ((*abs1).NumJour < (*abs2).NumJour)
        return -1; // absence 1 vient avant absence 2
    if ((*abs1).NumJour > (*abs2).NumJour)
        return 1; // absence 1 vient après absence 2
    // Si NumJour sont égaux, on compare par demi-journée
    return strcmp((*abs1).demiJour, (*abs2).demiJour);

    /*XXXXXXXXXXXXXXXXXXXXX Sous Fonction AfficherDetailAbsence
    XXXXXXXXXXXXXXXXXXXXXXXX*/
    void afficherDetailAbs(unsigned int totalAbs, int nbAbsParEtat[MAX_STATUT],
        unsigned int* compAbs, Absence tabAbs[MAXABS], int idEtu, unsigned int
        JourCourant) {
        //printf("attenteJ : %d \nnonJ : %d\nJ : %d\nattenteV : %d\n",
        attentejust, nonjusti, justi, attentevalid);
        if (nbAbsParEtat[ATTENTEJUSTIFICAT] > 0) {
            printf("- En attente justificatif\n");
            for (unsigned int j = 0; j < *compAbs; ++j) {
                if (tabAbs[j].etat == ATTENTEJUSTIFICAT && tabAbs[j].id ==
                idEtu && tabAbs[j].NumJour <= JourCourant) {
                    printf(" [%d] %u/%s\n", tabAbs[j].idAbs,
                    tabAbs[j].NumJour, tabAbs[j].demiJour);
                }
            }
        }
        if (nbAbsParEtat[ATTENTEVALIDATION] > 0) {
            printf("- En attente validation\n");
            for (unsigned int j = 0; j < *compAbs; ++j) {
                if (tabAbs[j].etat == ATTENTEVALIDATION && tabAbs[j].id ==
                idEtu && tabAbs[j].NumJour <= JourCourant) {
                    printf(" [%d] %u/%s", tabAbs[j].idAbs, tabAbs[j].NumJour,
                    tabAbs[j].demiJour);
                    if (tabAbs[j].explication[0] == VIDE && JourCourant >=
                    tabAbs[j].JourJusti) { //si une justification existe déjà pour cet
                    identifiant d'absence
                        printf(" (%s)\n", tabAbs[j].explication + 1);
                    }
                    else {
                        printf("\n"); //pour retour à la ligne si pas
                        d'explication antérieur à mettre
                    }
                }
            }
        }
        if (nbAbsParEtat[JUSTIFIEE] > 0) {
            printf("- Justifiees\n");
            for (unsigned int j = 0; j < *compAbs; ++j) {
                if (tabAbs[j].etat == JUSTIFIEE && tabAbs[j].id == idEtu &&
                tabAbs[j].NumJour <= JourCourant) {
```

```

        printf(" [%d] %u/%s", tabAbs[j].idAbs, tabAbs[j].NumJour,
tabAbs[j].demiJour);
        if (tabAbs[j].explication[0] == VIDE && JourCourant >=
tabAbs[j].JourJusti) { //si une justification existe déjà pour cet
identifiant d'absence
            printf(" (%s)\n", tabAbs[j].explication + 1);
        }
        else {
            printf("\n"); //pour retour à la ligne si pas
d'explication antérieur à mettre
        }
    }
}
}
if (nbAbsParEtat[NONJUSTIFIEE] > 0) {
    printf("- Non-justifiees\n");
    for (unsigned int j = 0; j < *compAbs; ++j) {
        if (tabAbs[j].etat == NONJUSTIFIEE && tabAbs[j].id == idEtu &&
tabAbs[j].NumJour <= JourCourant && strcmp(tabAbs[j].JustiAcceptee, "no")
== 0 && tabAbs[j].JourJusti <= JourCourant) {
            printf(" [%d] %u/%s", tabAbs[j].idAbs, tabAbs[j].NumJour,
tabAbs[j].demiJour);
            if (tabAbs[j].explication[0] == VIDE && JourCourant >=
tabAbs[j].JourJusti) { //si une justification existe déjà pour cet
identifiant d'absence
                printf(" (%s)\n", tabAbs[j].explication + 1);
            }
            else {
                printf("\n"); //pour retour à la ligne si pas
d'explication antérieur à mettre
            }
        }
    }
}
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Sous Fonction CopieDetabAbs
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
Absence* tabAbs2(unsigned int* compAbs, Absence* tabAbs) {
    Absence* absences2 = (Absence*)malloc(*compAbs * sizeof(Absence));
    if (absences2 == NULL) {
        printf("Erreur d'allocation mémoire\n");
        exit(1);
    }
    for (unsigned int i = 0; i < *compAbs; i++) {
        absences2[i] = tabAbs[i];
    }
    return absences2;
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Sous Fonction ChangeEtat
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
void changeEtat(Absence* absences2, unsigned int* compAbs, unsigned int
JourCourant, int idEtu) {
    for (unsigned int i = 0; i < *compAbs; i++) { //pour chaque etudiant
        if (absences2[i].id == idEtu) { //si idEtu est un identifiant d'un
etudiant inscrit
            if (absences2[i].NumJour <= JourCourant) {

```



```

        if (JourCourant - absences2[i].NumJour <= MAXNJ &&
absences2[i].JustiAcceptee[0] == VIDE && absences2[i].explication[0] ==
VIDE) {
            absences2[i].etat = ATTENTEJUSTIFICAT;
        }
        else if (absences2[i].etat == ATTENTEJUSTIFICAT &&
JourCourant - absences2[i].NumJour > MAXNJ) {
            absences2[i].etat = NONJUSTIFIEE;
        }
        else if (JourCourant - absences2[i].NumJour > MAXNJ &&
JourCourant < absences2[i].JourJusti) {
            absences2[i].etat = NONJUSTIFIEE;
        }
        else if (absences2[i].JourJusti > 0 && JourCourant <
absences2[i].JourJusti) {
            absences2[i].etat = ATTENTEJUSTIFICAT;
        }
        else if (absences2[i].JourJusti > 0 && JourCourant >=
absences2[i].JourJusti) {
            if (absences2[i].JourJusti - absences2[i].NumJour <=
MAXNJ) {
                if (absences2[i].JustiAcceptee[0] == VIDE)
                    absences2[i].etat = ATTENTEVALIDATION;
                else if (strcmp(absences2[i].JustiAcceptee, "ok")
== 0)
                    absences2[i].etat = JUSTIFIEE;
                else if (strcmp(absences2[i].JustiAcceptee, "ko")
== 0)
                    absences2[i].etat = NONJUSTIFIEE;
            }
            else
                absences2[i].etat = NONJUSTIFIEE;
        }
        else if (JourCourant - absences2[i].NumJour <= MAXNJ &&
strcmp(absences2[i].JustiAcceptee, "ko") == 0 && absences2[i].JourJusti >
JourCourant) {
            absences2[i].etat = ATTENTEJUSTIFICAT;
        }
        else if (absences2[i].JourJusti > 0 &&
absences2[i].JourJusti > JourCourant && absences2[i].etat ==
ATTENTEVALIDATION) {
            absences2[i].etat = ATTENTEJUSTIFICAT;
        }
        else if (JourCourant - absences2[i].NumJour <= MAXNJ &&
strcmp(absences2[i].JustiAcceptee, "ko") == 0 && absences2[i].JourJusti <=
JourCourant) {
            absences2[i].etat = NONJUSTIFIEE;
        }
    }
}
}
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction C7 Etudiant
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

```

```

void etudiant(Etudiant * tabEtu, unsigned int* compEtu, Absence * tabAbs,
unsigned int* compAbs) {
    int idEtu;
    unsigned int JourCourant;
    scanf("%d%u", &idEtu, &JourCourant);
    if (JourCourant < JOURCOMMENCE) {
        printf("Date incorrecte\n");
        return;
    }
    if (idEtu <= 0 || idEtu > *compEtu) {
        printf("Identifiant incorrect\n");
        return;
    }
    unsigned int totalAbs = 0;
    int nbAbsParEtat[MAX_ETAT] = { 0 };
    Absence* absences2 = tabAbs2(*compAbs, tabAbs);
    changeEtat(absences2, *compAbs, JourCourant, idEtu);
    for (int i = 0; i < *compAbs; i++) {
        if (absences2[i].id == idEtu) {
            if (absences2[i].NumJour <= JourCourant) {
                totalAbs += 1;
                nbAbsParEtat[absences2[i].etat] += 1;
            }
        }
    }
    printf("(%d) %s %d %d\n", tabEtu[idEtu - 1].id, tabEtu[idEtu -
1].NomEtu, tabEtu[idEtu - 1].NumGrp, totalAbs);
    if (totalAbs > 0) {
        qsort(absences2, *compAbs, sizeof(Absence), compare3);
        afficherDetailAbs(totalAbs, nbAbsParEtat, compAbs, absences2,
idEtu, JourCourant);
    }
}

```

b – Réussites

Notre plus grande réussite est certainement, la compréhension du sujet et des demandes émises par la constitution d'un interpréteur de commandes gérant les absences.

En effet, nous avons réussi à gérer ces inscriptions, ces absences en prenant en compte l'ensemble des conditions d'arrêts ou de l'affichage des messages d'erreur

explicités ou implicites dans le sujet. Ainsi, nos algorithmes réussis sont fonctionnels et répondent bel et bien au sujet.

c – Pistes d'améliorations

Tout d'abord, le principal point à améliorer dans notre code est sa factorisation. En effet, nos fonctions s'avèrent, malgré tous nos efforts, trop longues surtout la fonction « étudiant ». Ainsi, une piste d'amélioration est la création de fonctions auxiliaires, cité plus haut.

Une seconde piste d'amélioration est également de mieux optimiser nos structures de données afin de passer moins de paramètres à nos fonctions. Par exemple, nous pourrions directement ajouter les éléments de la structure Etudiant à la structure Absence. Ceci permettrait une meilleure lisibilité de notre programme, et la suppression de certains paramètres pour ainsi simplifier notre code.

Ensuite, nous aurions également pu gérer l'alignement des affichages afin de compléter notre code.

Et enfin, notre gestion du temps est un peu à améliorer, car nous pensons pouvoir réussir le sprint 4 complémentaire avec erreur en ajoutant les idées citées plus haut dans la fonction « étudiant ».

Ainsi, voici les principales pistes d'améliorations de notre programme :

- Création de fonctions auxiliaires
- Optimisation de nos structures de données
- Gérer l'alignement
- Gestion du temps : de la pile et des pointeurs

Outre ces grands axes, nous pensons avoir produit une réponse en cohérence avec les attendus du sujet.

ANNEXES

Code Source de notre Programme

```
#include <stdio.h>
#include <assert.h>
#include <string.h>
#include <stdlib.h> // pour qsort

#pragma warning (disable:4996)
enum { MAXNOM = 31, MAXETU = 100, MAXABS = 800 }; // nbre de caracteres du
nom, nbre d'étudiants, nbre d'absence max : 40jourmax x 2 (am et pm) x 100
(nbre etudiants) = 800
enum { MAXENTREE = 100 }; // nbre caracteres qu'on va écrire
enum { DEMI = 3 }; // am ou pm
enum { MAXCAR = 51 }; // nbre de caractere pour justificatif
enum { ACCEPTEE = 3 }; // ok ou ko
enum { MAXNJ = 3 }; //le nbre maximum de jour qui suit l'absence pour
justifié avant de ne plus être justifiable
enum Etat { NONJUSTIFIEE = 0, ATTENTEJUSTIFICAT = 1, ATTENTEVALIDATION = 2,
JUSTIFIEE = 3 }; //etat d'une absence
enum { VIDE = -1 }; //chaine vide
enum { NONTROUVEE = 0, TROUVEE = 1 };
enum { JOURCOMMENCE = 1, JOURFINI = 40 }; //num jour minimum est 1 et num
jour maximum de l'absence est 40
enum { ABSETU_DEFAILLANT = 5 }; //nbre d'absence max avant d'être
defaillant

/*XXXXXXXXXXXXXXXXXXXXX structure Etudiant XXXXXXXXXXXXXXXXXXXXX*/

typedef struct {
    int id;
    char NomEtu[MAXNOM];
    int NumGrp;//Numero de groupe de l'étudiant
} Etudiant;

/*XXXXXXXXXXXXXXXXXXXXX structure Absence XXXXXXXXXXXXXXXXXXXXX*/

typedef struct {
    int id; //id de l'étudiant
    unsigned int NumJour;
    char demiJour[DEMI];
    int idAbs; // identifiant de l'absence
    unsigned JourJusti;//Jour du dépôt de la justification
    char explication[MAXCAR]; // la justification par l'élève de l'absence
    unsigned int etat; //Etat de l'absence
```

```

    char JustiAcceptee[ACCEPTEE]; //décision du directeur : valider ou
    invalider une justification
} Absence;

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Inscription C1
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void inscription(Etudiant* tabEtu, unsigned int* compEtu);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Absence C2
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void absence(Absence* tabAbs, Etudiant* tabEtu, unsigned int* compAbs,
unsigned int* compEtu);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Sous Fonction Compare (trie Etudiant)
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

int compare(const void* a, const void* b);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Etudiants C3
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void etudiants(Etudiant* tabEtu, unsigned int* compEtu, Absence* tabAbs,
unsigned int* compAbs);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Justificatif C4
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void justificatif(Absence* tabAbs, unsigned int* compAbs);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Sous Fonction Compare2 (trie Etudiant +
Absence)
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

int compare2(const void* a, const void* b);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Validations C5
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void validations(Absence* tabAbs, unsigned int* compAbs, Etudiant* tabEtu,
unsigned int* compEtu);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Validation C6
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void validation(Absence* tabAbs, unsigned int* compAbs);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Sous Fonction Compare3 (trie Absence)
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

int compare3(const void* a, const void* b);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Etudiant C7
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void etudiant(Etudiant* tabEtu, unsigned int* compEtu, Absence* tabAbs,
unsigned int* compAbs);

```

```

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Defaillants C8
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void defaillants(Absence* tabAbs, unsigned int* compAbs, Etudiant* tabEtu,
unsigned int* compEtu);

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction MAIN
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

int main() {
    Etudiant etud[MAXETU];
    unsigned int compteurEtudiant = 0;
    Absence absences[MAXABS];
    unsigned int compteurAbsence = 0;
    char entrée[MAXENTREE];    //tableau car chaine de caractère
    while (1) {
        scanf("%s", entrée);
        if (strcmp(entrée, "inscription") == 0) {
            inscription(etud, &compteurEtudiant);
        }
        else if (strcmp(entrée, "absence") == 0) {
            absence(absences, etud, &compteurAbsence, &compteurEtudiant);
        }
        else if (strcmp(entrée, "etudiants") == 0) {
            etudiants(etud, &compteurEtudiant, absences, &compteurAbsence);
        }
        else if (strcmp(entrée, "justificatif") == 0) {
            justificatif(absences, &compteurAbsence);
        }
        else if (strcmp(entrée, "validations") == 0) {
            validations(absences, &compteurAbsence, etud,
&compteurEtudiant);
        }
        else if (strcmp(entrée, "validation") == 0) {
            validation(absences, &compteurAbsence);
        }
        else if (strcmp(entrée, "etudiant") == 0) {
            etudiant(etud, &compteurEtudiant, absences, &compteurAbsence);
        }
        else if (strcmp(entrée, "defaillants") == 0) {
            defaillants(absences, &compteurAbsence, etud,
&compteurEtudiant);
        }
        else if (strcmp(entrée, "exit") == 0) {
            return 0; //quitte le programme
        }
    }
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Inscription C1
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void inscription(Etudiant* tabEtu, unsigned int* compEtu) { // void car
fonction ne retourne rien
    char NomEtu[MAXNOM];
    int NumGrp;

```

```

    scanf("%s%d", NomEtu, &NumGrp); //pas de & pour NomEtu car tableau deja
    converti en pointeur
    //assert(strlen(NomEtu) <= 30)
    for (unsigned int i = 0; i < *compEtu; ++i) {
        if (strcmp(NomEtu, tabEtu[i].NomEtu) == 0 && NumGrp ==
tabEtu[i].NumGrp) {
            printf("Nom incorrect\n");
            return; // nom incorrect donc sort
        }
    }
    tabEtu[*compEtu].id = *compEtu + 1;
    strcpy(tabEtu[*compEtu].NomEtu, NomEtu);
    tabEtu[*compEtu].NumGrp = NumGrp;
    ++(*compEtu);
    printf("Inscription enregistree (%u)\n", *compEtu);
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Absence C2
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void absence(Absence* tabAbs, Etudiant* tabEtu, unsigned int* compAbs,
unsigned int* compEtu) {
    int id;
    unsigned int NumJour;
    char demiJour[DEMI];
    scanf("%d%u%s", &id, &NumJour, demiJour);
    for (unsigned int i = 0; i < *compAbs; ++i) {
        if (id == tabAbs[i].id && NumJour == tabAbs[i].NumJour &&
strcmp(demiJour, tabAbs[i].demiJour) == 0) { //si les entrées sont déjà
dans le tabAbs alors l'absence est connue
            printf("Absence deja connue\n");
            return;
        }
    }
    for (unsigned int j = 0; j < *compEtu; ++j) {
        if (id == tabEtu[j].id) { //si id de l'étudiant existe dans tabEtu
alors c un etudiant bien inscrit
            if (NumJour >= JOURCOMMENCE && NumJour <= JOURFINI) {
                if (strcmp(demiJour, "am") == 0 || (strcmp(demiJour, "pm")
== 0)) {
                    tabAbs[*compAbs].idAbs = *compAbs + 1; //on stock le
détail de l'absence entrées par l'utilisateur dans tabAbs
                    tabAbs[*compAbs].id = tabEtu[j].id;
                    tabAbs[*compAbs].NumJour = NumJour;
                    strcpy(tabAbs[*compAbs].demiJour, demiJour);
                    tabAbs[*compAbs].explication[0] = VIDE;
                    tabAbs[*compAbs].JustiAcceptee[0] = VIDE;
                    tabAbs[*compAbs].etat = ATTENTEJUSTIFICAT;
                    ++(*compAbs);
                    printf("Absence enregistree [%u]\n", *compAbs);
                    return;
                }
            }
            else {
                printf("Demi-journee incorrecte\n");
                return;
            }
        }
    }
    else {

```

```

        printf("Date incorrecte\n");
        return;
    }
}

printf("Identifiant incorrect\n");
return;
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Sous Fonction Compare
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

int compare(const void* a, const void* b) {
    //conversion
    const Etudiant* p1 = (const Etudiant*)a;
    const Etudiant* p2 = (const Etudiant*)b;
    if ((*p1).NumGrp < (*p2).NumGrp)
        return -1; // groupe de p1 est plus petit que groupe de p2
    if ((*p1).NumGrp > (*p2).NumGrp)
        return 1; // groupe de p1 est plus grand que groupe de p2
    // ici on sait que : groupe de p1 == groupe de p2, ils sont dans le
    même groupe
    return strcmp((*p1).NomEtu, (*p2).NomEtu); // on compare les 2 noms
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Etudiants C3
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void etudiants(Etudiant* tabEtu, unsigned int* compEtu, Absence* tabAbs,
unsigned int* compAbs) {
    unsigned int JourCourant;
    scanf("%u", &JourCourant);
    if (JourCourant < JOURCOMMENCE) { //si le JourCourant est inférieur à 1
        printf("Date incorrecte\n");
        return;
    }
    else if (*compEtu == 0) { //s'il n'y aucun etudiant inscrit à ce jour
        printf("Aucun inscrit\n");
        return;
    }
    else {
        qsort(tabEtu, *compEtu, sizeof(tabEtu[0]), compare); //trie
Etudiants
        for (unsigned int i = 0; i < *compEtu; ++i) { //pour chaque
etudiant
            unsigned int nbreAbs = 0;
            for (unsigned int j = 0; j < *compAbs; ++j) {
                if (tabAbs[j].NumJour <= JourCourant && tabAbs[j].id ==
tabEtu[i].id) {
                    nbreAbs++;
                }
            }
            printf("(%d) %s %d %d\n", tabEtu[i].id, tabEtu[i].NomEtu,
tabEtu[i].NumGrp, nbreAbs);
        }
    }
}

```



```

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Justificatif C4
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void justificatif(Absence* tabAbs, unsigned int* compAbs) {
    int idAbs;
    unsigned int JourJusti;
    char explication[MAXCAR];
    scanf("%u%u%[^\\n]", &idAbs, &JourJusti, explication); //%^\\n pour
inclure les espaces de l'explication
    for (unsigned int i = 0; i < *compAbs; ++i) {
        if (tabAbs[i].idAbs == idAbs) { // l'idAbs connu
            if (JourJusti >= JOURCOMMENCE && JourJusti >=
tabAbs[i].NumJour) { // si numjour supérieur ou égale à 1 et numjour après
la date absence ou à la date d'absence
                //printf("entre"); pour voir si ça rentre dans la condition
                if (tabAbs[i].explication[0] == VIDE) { // explication est
vide donc l'absence pas encore justifiée
                    tabAbs[i].JourJusti = JourJusti; // Enregistrer le
numéro de jour de la justification
                    strcpy(tabAbs[i].explication, explication); // copie
l'explication dans Enregistrer l'explication
                    printf("Justificatif enregistré\\n");
                    if (JourJusti - tabAbs[i].NumJour > MAXNJ) { // si
justificatif absence + de trois jours de l'absence //nbre magique
                        tabAbs[i].etat = NONJUSTIFIEE; // non-justifiée
                        //printf("%d\\n", tabAbs[i].etat);
                        return;
                    }
                }
                else {
                    tabAbs[i].etat = ATTENTEVALIDATION; // en cours de
traitement
                    //printf("%d\\n", tabAbs[i].etat);
                    return;
                }
            }
            else {
                printf("Justificatif déjà connu\\n"); //si une
justification existe déjà pour cet identifiant d'absence
                return;
            }
        }
        else {
            printf("Date incorrecte\\n");
            return;
        }
    }
    printf("Identifiant incorrect\\n");
    return;
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Sous Fonction Compare2
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

int compare2(const void* a, const void* b) {
    //conversion

```

```

    const Absence* p1 = (const Absence*)a;
    const Absence* p2 = (const Absence*)b;
    if ((*p1).id < (*p2).id) //trier par numero d'etudiant croissant
        return -1; // id de p1 est plus petit que id de p2
    if ((*p1).id > (*p2).id)
        return 1; // id de p1 est plus grand que id de p2
    // Si les id sont égaux : meme etudiant, comparaison par NumJour
    (chronologie)
    if ((*p1).NumJour < (*p2).NumJour)
        return -1; // absence de p1 vient avant absence de p2
    if ((*p1).NumJour > (*p2).NumJour)
        return 1; // absence de p1 vient après celle de p2
    // Si les id et NumJour sont égaux, on compare par demi-journée
    return strcmp((*p1).demiJour, (*p2).demiJour);
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Validations C5
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void validations(Absence* tabAbs, unsigned int* compAbs, Etudiant* tabEtu,
unsigned int* compEtu) {
    unsigned int validationTrouvee = NONTROUVEE; //si validation trouvé
    qsort(tabAbs, *compAbs, sizeof(tabAbs[0]), compare2);
    for (unsigned i = 0; i < *compAbs; ++i) {
        if (tabAbs[i].etat == ATTENTEVALIDATION) { // si l'absence est en
cours de traitement
            for (unsigned int j = 0; j < *compEtu; ++j) {
                if (tabAbs[i].id == tabEtu[j].id) { //jointure par l'id
                    printf("[%d] (%d) %s %d %u/%s (%s)\n", tabAbs[i].idAbs,
tabAbs[i].id, tabEtu[j].NomEtu, tabEtu[j].NumGrp, tabAbs[i].NumJour,
tabAbs[i].demiJour, tabAbs[i].explication + 1); // tabJusti[i].explication
+ 1 car on ne veut pas d'espace entre ( et l'explication
                    validationTrouvee = TROUVEE; // une absence en attente
de validation a ete trouvee
                }
            }
        }
    }
    if (validationTrouvee == NONTROUVEE) { // Si aucune absence en attente
de validation n'a ete trouvee
        printf("Aucune validation en attente\n");
        return;
    }
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Validation C6
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void validation(Absence* tabAbs, unsigned int* compAbs) {
    int idAbsence;
    char JustiAcceptee[ACCEPTEE];
    scanf("%d%s", &idAbsence, JustiAcceptee);
    for (unsigned int i = 0; i < *compAbs; ++i) {
        if (tabAbs[i].idAbs == idAbsence) { //si l'idAbsence est une
absence connue et en attente de validation
            //printf("%d\n", tabAbs[i].etat); //verifie si l'etat est en
attente pour voir s'il rentre dans la condition
            if (tabAbs[i].JustiAcceptee[0] == VIDE) {

```

```

        //printf("%d\n", tabAbs[i].etat);
        if (tabAbs[i].etat == ATTENTEVALIDATION) {
            if (strcmp(JustiAcceptee, "ok") == 0) {
                strcpy(tabAbs[i].JustiAcceptee, JustiAcceptee);
//copie caractères de JustiAcceptee dans tabAbs
                tabAbs[i].etat = JUSTIFIEE; // l'absence est
justifiee

                printf("Validation enregistree\n");
                return;
            }
            else if (strcmp(JustiAcceptee, "ko") == 0) {
                strcpy(tabAbs[i].JustiAcceptee, JustiAcceptee);
                tabAbs[i].etat = NONJUSTIFIEE; // l'absence est
non-justifiee

                printf("Validation enregistree\n");
                return;
            }
            else { // le cas où (strcmp(JustiAcceptee, "ok") != 0
&& strcmp(JustiAcceptee, "ko" != 0))
                printf("Code incorrect\n");
                return;
            }
        }
        else {
            printf("Identifiant incorrect\n"); //si autre etat
qu'en attente de validation
            return;
        }
    }
    else {
        printf("Validation deja connue\n");
        return;
    }
}

printf("Identifiant incorrect\n");
return;
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Sous Fonction Compare3
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

int compare3(const void* a, const void* b) {
    //conversion
    const Absence* abs1 = (const Absence*)a;
    const Absence* abs2 = (const Absence*)b;
    if ((*abs1).NumJour < (*abs2).NumJour)
        return -1; // absence 1 vient avant absence 2
    if ((*abs1).NumJour > (*abs2).NumJour)
        return 1; // absence 1 vient après absence 2
    // Si NumJour sont égaux, on compare par demi-journée
    return strcmp((*abs1).demiJour, (*abs2).demiJour);
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Etudiant C7
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

```

```

void etudiant(Etudiant* tabEtu, unsigned int* compEtu, Absence* tabAbs,
unsigned int* compAbs) {
    int idEtu;
    unsigned int JourCourant;
    int nonjusti = 0;
    int attentejust = 0;
    int justi = 0;
    int attentevalid = 0;
    scanf("%d%u", &idEtu, &JourCourant);
    if (JourCourant < JOURCOMMENCE) {
        printf("Date incorrecte\n");
        return;
    }
    qsort(tabAbs, *compAbs, sizeof(tabAbs[0]), compare3); //on trie par
    absence de l'étudiant
    for (unsigned int i = 0; i < *compEtu; ++i) { //pour chaque etudiant
        if (tabEtu[i].id == idEtu) { //si idEtu est un identifiant d'un
            etudiant inscrit
                unsigned int nbreAbs = 0;
                for (unsigned int j = 0; j < *compAbs; ++j) {
                    if (tabAbs[j].NumJour <= JourCourant && tabAbs[j].id ==
tabEtu[i].id) {
                        nbreAbs++;
                        //printf("idAbs : %d etat : %d", tabAbs[j].id,
tabAbs[j].etat);
                        //printf("(%d) etat = %d\n", tabAbs[j].idAbs,
tabAbs[j].etat);
                        //printf("%d", tabAbs[j].etat);

                        if (JourCourant - tabAbs[j].NumJour <= MAXNJ &&
tabAbs[j].JustiAcceptee == VIDE && tabAbs[j].explication[0] == VIDE)
{ //l'absence ne dépasse pas 3 jours sans justificat, elle est donc en
attente de justification
                            tabAbs[j].etat = ATTENTEJUSTIFICAT;
                            //printf("la");
                        }
                        else if (tabAbs[j].etat == ATTENTEJUSTIFICAT &&
JourCourant - tabAbs[j].NumJour > MAXNJ) { // ATTENTEJUSTIF JOUR ABSENCE >
3 JOUR
                                    tabAbs[j].etat = NONJUSTIFIEE;
                                    //printf("go");
                                }
                                else if (JourCourant - tabAbs[j].NumJour > MAXNJ &&
JourCourant < tabAbs[j].JourJusti) {
                                    tabAbs[j].etat = NONJUSTIFIEE;
                                }
                                else if (JourCourant < tabAbs[j].JourJusti &&
tabAbs[j].explication[0] != VIDE) { // y a une justification après le
JourCourant
                                    tabAbs[j].etat = ATTENTEJUSTIFICAT;
                                    //printf("lo");
                                }
                                else if (JourCourant >= tabAbs[j].JourJusti &&
tabAbs[i].explication[0] != VIDE) { // y a une justification avant le
JourCourant
                                    if (tabAbs[j].JourJusti - tabAbs[j].NumJour <=
MAXNJ) {
                                        if (tabAbs[j].JustiAcceptee[0] == VIDE)

```

```

        tabAbs[j].etat = ATTENTEVALIDATION;
    else if (strcmp(tabAbs[j].JustiAcceptee, "ok")
== 0)
        tabAbs[j].etat = JUSTIFIEE;
    else if (strcmp(tabAbs[j].JustiAcceptee, "ko")
== 0)
        tabAbs[j].etat = NONJUSTIFIEE;
    }
    else
        tabAbs[j].etat = NONJUSTIFIEE;
}
else if (JourCourant - tabAbs[j].NumJour <= MAXNJ &&
tabAbs[j].etat == NONJUSTIFIEE && tabAbs[i].JustiAcceptee == VIDE) {
    tabAbs[j].etat = ATTENTEJUSTIFICAT;
    //printf("hello");
}
else if (JourCourant - tabAbs[j].NumJour <= MAXNJ &&
strcmp(tabAbs[i].JustiAcceptee, "ko") == 0 && tabAbs[i].JourJusti >
JourCourant) {
    tabAbs[j].etat = ATTENTEJUSTIFICAT;
    //printf("po");
}

else if (tabAbs[j].JourJusti > JourCourant &&
tabAbs[j].etat == ATTENTEVALIDATION) {
    tabAbs[j].etat = ATTENTEJUSTIFICAT;
    //printf("pa");
}
if (tabAbs[j].etat == ATTENTEJUSTIFICAT) {
    attentejust++;
}
else if (tabAbs[j].etat == NONJUSTIFIEE) {
    nonjusti++;
}
else if (tabAbs[j].etat == JUSTIFIEE) {
    justii++;
}
else {
    attentevalid++;
}
}
}
//printf("attenteJ : %d \nnonJ : %d\nJ : %d\nattenteV : %d\n",
attentejust, nonjusti, justii, attentevalid);
printf("(%d) %s %d %d\n", tabEtu[i].id, tabEtu[i].NomEtu,
tabEtu[i].NumGrp, nbreAbs);
if (attentejust > 0) {
    printf("- En attente justificatif\n");
    for (unsigned int j = 0; j < *compAbs; ++j) {
        if (tabAbs[j].etat == ATTENTEJUSTIFICAT && tabAbs[j].id
== tabEtu[i].id && tabAbs[j].NumJour <= JourCourant) {
            printf(" [%d] %u/%s\n", tabAbs[j].idAbs,
tabAbs[j].NumJour, tabAbs[j].demiJour);
            --attentejust;
        }
    }
}
if (attentevalid > 0) {

```

```

printf("- En attente validation\n");
for (unsigned int j = 0; j < *compAbs; ++j) {
    if (tabAbs[j].etat == ATTENTEVALIDATION && tabAbs[j].id
== tabEtu[i].id && tabAbs[j].NumJour <= JourCourant) {
        printf(" [%d] %u/%s", tabAbs[j].idAbs,
tabAbs[j].NumJour, tabAbs[j].demiJour);
        --attentevalid;
        if (tabAbs[j].explication[0] != VIDE &&
JourCourant >= tabAbs[j].JourJusti) { //si une justification existe déjà
pour cet identifiant d'absence
            printf(" (%s)\n", tabAbs[j].explication + 1);
        }
        else {
            printf("\n"); //pour retour à la ligne si pas
d'explication antérieur à mettre
        }
    }
}
if (justi > 0) {
    printf("- Justifiees\n");
    for (unsigned int j = 0; j < *compAbs; ++j) {
        if (tabAbs[j].etat == JUSTIFIEE && tabAbs[j].id ==
tabEtu[i].id && tabAbs[j].NumJour <= JourCourant) {
            printf(" [%d] %u/%s", tabAbs[j].idAbs,
tabAbs[j].NumJour, tabAbs[j].demiJour);
            --justi;
            if (tabAbs[j].explication[0] != VIDE &&
JourCourant >= tabAbs[j].JourJusti) { //si une justification existe déjà
pour cet identifiant d'absence
                printf(" (%s)\n", tabAbs[j].explication + 1);
            }
            else {
                printf("\n"); //pour retour à la ligne si pas
d'explication antérieur à mettre
            }
        }
    }
}
if (nonjusti > 0) {
    printf("- Non-justifiees\n");
    for (unsigned int j = 0; j < *compAbs; ++j) {
        if (tabAbs[j].etat == NONJUSTIFIEE && tabAbs[j].id ==
tabEtu[i].id && tabAbs[j].NumJour <= JourCourant) {
            printf(" [%d] %u/%s", tabAbs[j].idAbs,
tabAbs[j].NumJour, tabAbs[j].demiJour);
            --nonjusti;
            if (tabAbs[j].explication[0] != VIDE &&
JourCourant >= tabAbs[j].JourJusti) { //si une justification existe déjà
pour cet identifiant d'absence
                printf(" (%s)\n", tabAbs[j].explication + 1);
            }
            else {
                printf("\n"); //pour retour à la ligne si pas
d'explication antérieur à mettre
            }
        }
    }
}
}

```

```

    }
    return; //plus d'absence donc sort
}
}
printf("Identifiant incorrect\n");
return;
}

/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Fonction Defaillants C8
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/

void defaillants(Absence* tabAbs, unsigned int* compAbs, Etudiant* tabEtu,
unsigned int* compEtu) {
    int JourCourant;
    unsigned int defaillantTrouvee = NONTROUVEE;
    scanf("%d", &JourCourant);
    qsort(tabEtu, *compEtu, sizeof(tabEtu[0]), compare);
    if (JourCourant < JOURCOMMENCE) {
        printf("Date incorrecte\n");
        return;
    }
    else {
        for (unsigned int i = 0; i < *compEtu; ++i) {
            unsigned int compAbsEtu = 0; //compte absence pour chaque
            etudiant individuellement
            unsigned int nbJNonJusti = 0; //compte absence non justifiée
            d'un etudiant
            for (unsigned int j = 0; j < *compAbs; ++j) {
                if (tabEtu[i].id == tabAbs[j].id && tabAbs[j].NumJour <=
                JourCourant) {
                    if (tabAbs[j].etat == ATTENTEJUSTIFICAT && JourCourant
                    - tabAbs[j].NumJour > MAXNJ) {
                        //tabAbs[j].etat = NONJUSTIFIEE;
                        ++nbJNonJusti;
                    }
                    //else if (JourCourant - tabAbs[j].NumJour <= MAXNJ) {
                    //tabAbs[j].etat = ATTENTEJUSTIFICAT;
                    //}
                    if (tabAbs[j].etat == NONJUSTIFIEE) {
                        ++nbJNonJusti;
                    }
                    ++compAbsEtu;
                }
            }
            if (nbJNonJusti >= ABSETU_DEFAILLANT) { //étudiant a plus de 5
            absences non-justifiées à ce jour
                printf("(%d) %s %d %u\n", tabEtu[i].id, tabEtu[i].NomEtu,
                tabEtu[i].NumGrp, compAbsEtu);
                defaillantTrouvee = TROUVEE;
            }
        }
    }
    if (defaillantTrouvee == NONTROUVEE) {
        printf("Aucun defaillant\n");
        return; // aucun étudiant defaillant trouvé sort
    }
}

```

Trace d'exécution du test du sprint (3) de plus haut niveau atteint :

Sprint 3 – Sans Erreur : (in MOODLE)

etudiants 1

Aucun inscrit

validations

Aucune validation en attente

validations

Aucune validation en attente

etudiants 1

Aucun inscrit

inscription tomos 202

Inscription enregistree (1)

inscription cailin 202

Inscription enregistree (2)

inscription heino 40

Inscription enregistree (3)

inscription dunja 101

Inscription enregistree (4)

inscription severo 50

Inscription enregistree (5)

inscription raina 50

Inscription enregistree (6)

absence 6 10 pm

Absence enregistree [1]

absence 3 15 am

Absence enregistree [2]

absence 2 16 pm

Absence enregistree [3]

XU-108 XU-110

absence 3 30 pm

Absence enregistree [4]

absence 6 35 am

Absence enregistree [5]

inscription leslie 101

Inscription enregistree (7)

absence 5 25 pm

Absence enregistree [6]

absence 5 22 pm

Absence enregistree [7]

absence 3 3 pm

Absence enregistree [8]

etudiant 3 42

(3) heino 40 3

- Non-justifiees

[8] 3/pm

[2] 15/am

[4] 30/pm

absence 2 36 am

Absence enregistree [9]

inscription kelda 50

Inscription enregistree (8)

inscription anselma 103

Inscription enregistree (9)

absence 9 13 am

Absence enregistree [10]

etudiant 3 27

(3) heino 40 2

- Non-justifiees

[8] 3/pm

XU-108 XU-110

[2] 15/am

absence 7 16 pm

Absence enregistree [11]

absence 1 13 am

Absence enregistree [12]

absence 3 24 am

Absence enregistree [13]

inscription rebeccah 201

Inscription enregistree (10)

absence 8 5 am

Absence enregistree [14]

absence 4 38 pm

Absence enregistree [15]

absence 7 12 pm

Absence enregistree [16]

absence 3 21 pm

Absence enregistree [17]

absence 8 8 am

Absence enregistree [18]

etudiants 28

(3) heino 40 4

(8) kelda 50 2

(6) raina 50 1

(5) severo 50 2

(4) dunja 101 0

(7) leslie 101 2

(9) anselma 103 1

(10) rebeccah 201 0

(2) cailin 202 1

(1) tomos 202 1

XU-108 XU-110

justificatif 5 40 convoc. tribunal

Justificatif enregistre

absence 3 6 pm

Absence enregistree [19]

absence 9 4 am

Absence enregistree [20]

absence 5 38 am

Absence enregistree [21]

absence 4 40 am

Absence enregistree [22]

absence 10 39 am

Absence enregistree [23]

etudiant 5 35

(5) severo 50 2

- Non-justifiees

[7] 22/pm

[6] 25/pm

etudiant 9 5

(9) anselma 103 1

- En attente justificatif

[20] 4/am

etudiant 3 14

(3) heino 40 2

- Non-justifiees

[8] 3/pm

[19] 6/pm

absence 1 6 am

Absence enregistree [24]

absence 9 18 am

Absence enregistree [25]

XU-108 XU-110

absence 9 26 am

Absence enregistree [26]

justificatif 11 18 certif. medical

Justificatif enregistre

absence 8 6 am

Absence enregistree [27]

absence 5 30 am

Absence enregistree [28]

absence 10 6 am

Absence enregistree [29]

absence 2 35 pm

Absence enregistree [30]

justificatif 6 26 convoc. tribunal

Justificatif enregistre

absence 9 23 pm

Absence enregistree [31]

justificatif 17 24 permis de conduire

Justificatif enregistre

validation 11 ko

Validation enregistree

etudiant 6 21

(6) raina 50 1

- Non-justifiees

[1] 10/pm

absence 9 3 am

Absence enregistree [32]

absence 4 2 am

Absence enregistree [33]

justificatif 1 11 permis de conduire

Justificatif enregistre

XU-108 XU-110

absence 10 22 am

Absence enregistrée [34]

justificatif 7 23 certif. medical

Justificatif enregistré

validations

[17] (3) heino 40 21/pm (permis de conduire)

[7] (5) severo 50 22/pm (certif. medical)

[6] (5) severo 50 25/pm (convoc. tribunal)

[1] (6) raina 50 10/pm (permis de conduire)

justificatif 30 39 certif. medical

Justificatif enregistré

absence 7 34 pm

Absence enregistrée [35]

justificatif 29 12 j'aime pas ce prof

Justificatif enregistré

justificatif 15 41 j'avais d'autres choses à faire

Justificatif enregistré

absence 2 32 pm

Absence enregistrée [36]

justificatif 2 21 permis de conduire

Justificatif enregistré

etudiants 26

(3) heino 40 5

(8) kelda 50 3

(6) raina 50 1

(5) severo 50 2

(4) dunja 101 1

(7) leslie 101 2

(9) anselma 103 6

(10) rebeccah 201 2

XU-108 XU-110

(2) cailin 202 1

(1) tomos 202 2

validation 6 ko

Validation enregistree

absence 7 1 am

Absence enregistree [37]

justificatif 8 7 j'avais piscine

Justificatif enregistre

absence 1 33 am

Absence enregistree [38]

absence 6 39 am

Absence enregistree [39]

absence 8 13 pm

Absence enregistree [40]

absence 3 4 am

Absence enregistree [41]

validations

[17] (3) heino 40 21/pm (permis de conduire)

[15] (4) dunja 101 38/pm (j'avais d'autres choses a faire)

[7] (5) severo 50 22/pm (certif. medical)

[1] (6) raina 50 10/pm (permis de conduire)

absence 9 33 am

Absence enregistree [42]

justificatif 37 6 permis de conduire

Justificatif enregistre

etudiant 2 20

(2) cailin 202 1

- Non-justifiees

[3] 16/pm

absence 3 10 am

XU-108 XU-110

Absence enregistrée [43]

justificatif 21 42 j'avais d'autres choses à faire

Justificatif enregistré

justificatif 31 26 convoc. tribunal

Justificatif enregistré

absence 8 6 pm

Absence enregistrée [44]

justificatif 19 9 j'avais d'autres choses à faire

Justificatif enregistré

absence 1 28 pm

Absence enregistrée [45]

absence 9 24 pm

Absence enregistrée [46]

justificatif 46 28 convoc. tribunal

Justificatif enregistré

absence 4 30 am

Absence enregistrée [47]

justificatif 18 12 ma chaîne de vélo a dérailé

Justificatif enregistré

justificatif 16 13 j'avais piscine

Justificatif enregistré

justificatif 42 36 convoc. tribunal

Justificatif enregistré

justificatif 45 30 j'aime pas ce cours

Justificatif enregistré

absence 9 23 am

Absence enregistrée [48]

justificatif 22 45 ma chaîne de vélo a dérailé

Justificatif enregistré

justificatif 39 40 permis de conduire

XU-108 XU-110

Justificatif enregistré

validation 39 ko

Validation enregistrée

absence 2 3 pm

Absence enregistrée [49]

validation 17 ok

Validation enregistrée

justificatif 47 32 Journée défense et citoyenneté

Justificatif enregistré

validation 15 ok

Validation enregistrée

absence 7 39 pm

Absence enregistrée [50]

validation 45 ko

Validation enregistrée

justificatif 50 45 j'avais d'autres choses à faire

Justificatif enregistré

absence 8 1 pm

Absence enregistrée [51]

justificatif 51 5 certif. médical

Justificatif enregistré

absence 5 11 am

Absence enregistrée [52]

justificatif 49 8 Journée défense et citoyenneté

Justificatif enregistré

absence 7 24 am

Absence enregistrée [53]

absence 1 26 pm

Absence enregistrée [54]

justificatif 44 10 j'aime pas ce cours

XU-108 XU-110

Justificatif enregistré

validation 16 ok

Validation enregistrée

justificatif 53 26 certif. medical

Justificatif enregistré

validation 1 ok

Validation enregistrée

étudiant 4 45

(4) dunja 101 4

- En attente validation

[47] 30/am (Journée défense et citoyenneté)

- Justifiées

[15] 38/pm (j'avais d'autres choses à faire)

- Non-justifiées

[33] 2/am

[22] 40/am (ma chaîne de vélo a dérailé)

étudiants 8

(3) heino 40 3

(8) kelda 50 5

(6) raina 50 0

(5) severo 50 0

(4) dunja 101 1

(7) leslie 101 1

(9) anselma 103 2

(10) rebeccah 201 1

(2) cailin 202 1

(1) tomos 202 1

justificatif 36 35 ma chaîne de vélo a dérailé

Justificatif enregistré

validation 47 ko

XU-108 XU-110

Validation enregistree

absence 4 36 pm

Absence enregistree [55]

justificatif 28 33 convoc. tribunal

Justificatif enregistre

absence 2 25 pm

Absence enregistree [56]

justificatif 56 30 convoc. tribunal

Justificatif enregistre

validation 36 ko

Validation enregistree

validation 19 ko

Validation enregistree

absence 8 9 pm

Absence enregistree [57]

validation 53 ok

Validation enregistree

validation 28 ok

Validation enregistree

absence 3 28 am

Absence enregistree [58]

absence 5 23 pm

Absence enregistree [59]

justificatif 58 31 certif. medical

Justificatif enregistre

justificatif 59 27 Journee defense et citoyennete

Justificatif enregistre

absence 4 16 am

Absence enregistree [60]

absence 1 15 pm

XU-108 XU-110

Absence enregistrée [61]

justificatif 60 17 Journée défense et citoyenneté

Justificatif enregistré

validation 7 ko

Validation enregistrée

validation 58 ok

Validation enregistrée

justificatif 61 16 j'aime pas ce cours

Justificatif enregistré

validation 42 ok

Validation enregistrée

absence 4 31 am

Absence enregistrée [62]

justificatif 38 34 permis de conduire

Justificatif enregistré

validation 31 ko

Validation enregistrée

justificatif 26 30 j'avais piscine

Justificatif enregistré

justificatif 41 10 j'aime pas ce prof

Justificatif enregistré

validation 61 ok

Validation enregistrée

absence 4 29 pm

Absence enregistrée [63]

absence 4 6 pm

Absence enregistrée [64]

justificatif 54 31 ma chaîne de vélo a dérailé

Justificatif enregistré

absence 1 16 pm

XU-108 XU-110

Absence enregistrée [65]

justificatif 65 19 ma chaîne de vélo a dérailé

Justificatif enregistré

absence 1 4 pm

Absence enregistrée [66]

justificatif 55 39 convoc. police

Justificatif enregistré

justificatif 48 25 certif. médical

Justificatif enregistré

validation 48 ko

Validation enregistrée

validation 65 ko

Validation enregistrée

absence 9 26 pm

Absence enregistrée [67]

justificatif 67 30 convoc. tribunal

Justificatif enregistré

validation 55 ok

Validation enregistrée

absence 9 33 pm

Absence enregistrée [68]

justificatif 68 38 j'aime pas ce prof

Justificatif enregistré

validation 60 ok

Validation enregistrée

validation 38 ko

Validation enregistrée

absence 1 31 am

Absence enregistrée [69]

justificatif 69 34 j'aime pas ce prof

XU-108 XU-110

Justificatif enregistré

justificatif 66 6 j'avais d'autres choses à faire

Justificatif enregistré

validation 69 ok

Validation enregistrée

validation 66 ko

Validation enregistrée

étudiants 1

(3) heino 40 0

(8) kelda 50 1

(6) raina 50 0

(5) severo 50 0

(4) dunja 101 0

(7) leslie 101 1

(9) anselma 103 0

(10) rebeccah 201 0

(2) cailin 202 0

(1) tomos 202 0

étudiants 41

(3) heino 40 9

(8) kelda 50 7

(6) raina 50 3

(5) severo 50 6

(4) dunja 101 9

(7) leslie 101 6

(9) anselma 103 11

(10) rebeccah 201 3

(2) cailin 202 6

(1) tomos 202 9

validations

XU-108 XU-110

Aucune validation en attente

etudiant 1 1

(1) tomos 202 0

etudiant 1 45

(1) tomos 202 9

- Justifiees

[61] 15/pm (j'aime pas ce cours)

[69] 31/am (j'aime pas ce prof)

- Non-justifiees

[66] 4/pm (j'avais d'autres choses a faire)

[24] 6/am

[12] 13/am

[65] 16/pm (ma chaine de velo a deraille)

[54] 26/pm (ma chaine de velo a deraille)

[45] 28/pm (j'aime pas ce cours)

[38] 33/am (permis de conduire)

etudiant 2 1

(2) cailin 202 0

etudiant 2 45

(2) cailin 202 6

- Non-justifiees

[49] 3/pm (Journee defense et citoyennete)

[3] 16/pm

[56] 25/pm (convoc. tribunal)

[36] 32/pm (ma chaine de velo a deraille)

[30] 35/pm (certif. medical)

[9] 36/am

etudiant 3 1

(3) heino 40 0

etudiant 3 45

XU-108 XU-110

(3) heino 40 9

- Justifiees

[17] 21/pm (permis de conduire)

[58] 28/am (certif. medical)

- Non-justifiees

[8] 3/pm (j'avais piscine)

[41] 4/am (j'aime pas ce prof)

[19] 6/pm (j'avais d'autres choses a faire)

[43] 10/am

[2] 15/am (permis de conduire)

[13] 24/am

[4] 30/pm

etudiant 4 1

(4) dunja 101 0

etudiant 4 45

(4) dunja 101 9

- Justifiees

[60] 16/am (Journee defense et citoyennete)

[55] 36/pm (convoc. police)

[15] 38/pm (j'avais d'autres choses a faire)

- Non-justifiees

[33] 2/am

[64] 6/pm

[63] 29/pm

[47] 30/am (Journee defense et citoyennete)

[62] 31/am

[22] 40/am (ma chaine de velo a deraille)

etudiant 5 1

(5) severo 50 0

etudiant 5 45

XU-108 XU-110

(5) severo 50 6

- Justifiees

[28] 30/am (convoc. tribunal)

- Non-justifiees

[52] 11/am

[7] 22/pm (certif. medical)

[59] 23/pm (Journée défense et citoyenneté)

[6] 25/pm (convoc. tribunal)

[21] 38/am (j'avais d'autres choses à faire)

etudiant 6 1

(6) raina 50 0

etudiant 6 45

(6) raina 50 3

- Justifiees

[1] 10/pm (permis de conduire)

- Non-justifiees

[5] 35/am (convoc. tribunal)

[39] 39/am (permis de conduire)

etudiant 7 1

(7) leslie 101 1

- En attente justificatif

[37] 1/am

etudiant 7 45

(7) leslie 101 6

- Justifiees

[16] 12/pm (j'avais piscine)

[53] 24/am (certif. medical)

- Non-justifiees

[37] 1/am (permis de conduire)

[11] 16/pm (certif. medical)

XU-108 XU-110

[35] 34/pm

[50] 39/pm (j'avais d'autres choses a faire)

etudiant 8 1

(8) kelda 50 1

- En attente justificatif

[51] 1/pm

etudiant 8 45

(8) kelda 50 7

- Non-justifiees

[51] 1/pm (certif. medical)

[14] 5/am

[27] 6/am

[44] 6/pm (j'aime pas ce cours)

[18] 8/am (ma chaine de velo a deraille)

[57] 9/pm

[40] 13/pm

etudiant 9 1

(9) anselma 103 0

etudiant 9 45

(9) anselma 103 11

- Justifiees

[42] 33/am (convoc. tribunal)

- Non-justifiees

[32] 3/am

[20] 4/am

[10] 13/am

[25] 18/am

[48] 23/am (certif. medical)

[31] 23/pm (convoc. tribunal)

[46] 24/pm (convoc. tribunal)

XU-108 XU-110

[26] 26/am (j'avais piscine)

[67] 26/pm (convoc. tribunal)

[68] 33/pm (j'aime pas ce prof)

etudiant 10 1

(10) rebeccah 201 0

etudiant 10 45

(10) rebeccah 201 3

- Non-justifiees

[29] 6/am (j'aime pas ce prof)

[34] 22/am

[23] 39/am

exit

C:\Users\sarah\OneDrive\Bureau\C\x64\Debug\C.exe (processus 30412) s'est terminé avec le code 0 (0x0).

Appuyez sur une touche pour fermer cette fenêtre. . .

Sprint 3 – Avec Erreurs : (in MOODLE)

etudiants 1

Aucun inscrit

validations

Aucune validation en attente

inscription hildegard 202

Inscription enregistree (1)

inscription hildegard 202

Nom incorrect

inscription christer 203

Inscription enregistree (2)

absence 1 8 pm

Absence enregistree [1]

XU-108 XU-110

inscription rebeccah 101

Inscription enregistree (3)

inscription dakota 30

Inscription enregistree (4)

absence 101 1 pm

Identifiant incorrect

absence 2 17 am

Absence enregistree [2]

absence 3 25 am

Absence enregistree [3]

etudiant 2 10

(2) christer 203 0

inscription anselma 203

Inscription enregistree (5)

absence 1 36 pm

Absence enregistree [4]

absence 1 41 pm

Date incorrecte

validations

Aucune validation en attente

justificatif 0 10 j'ai mal au ventre

Identifiant incorrect

inscription nicodemus 40

Inscription enregistree (6)

absence 1 34 pm

Absence enregistree [5]

inscription christer 203

Nom incorrect

inscription giselle 101

Inscription enregistree (7)

XU-108 XU-110

absence 1 17 am

Absence enregistree [6]

absence 5 24 am

Absence enregistree [7]

absence 7 3 pm

Absence enregistree [8]

absence 1 34 pm

Absence deja connue

absence 7 30 pm

Absence enregistree [9]

absence 1 3 am

Absence enregistree [10]

inscription salomea 203

Inscription enregistree (8)

absence 5 30 pm

Absence enregistree [11]

justificatif 6 20 ma chaine de velo a deraille

Justificatif enregistre

absence 2 27 am

Absence enregistree [12]

absence 6 29 am

Absence enregistree [13]

etudiants 23

(4) dakota 30 0

(6) nicodemus 40 0

(7) giselle 101 1

(3) rebeccah 101 0

(1) hildegard 202 3

(5) anselma 203 0

(2) christer 203 1

XU-108 XU-110

(8) salomea 203 0

absence 1 1 an

Demi-journee incorrecte

absence 2 29 pm

Absence enregistree [14]

inscription savino 101

Inscription enregistree (9)

absence 1 6 pm

Absence enregistree [15]

absence 2 21 pm

Absence enregistree [16]

absence 3 13 am

Absence enregistree [17]

absence 2 13 am

Absence enregistree [18]

etudiant 3 45

(3) rebeccah 101 2

- Non-justifiees

[17] 13/am

[3] 25/am

justificatif 8 6 convoc. tribunal

Justificatif enregistre

justificatif 9 29 convoc. tribunal

Date incorrecte

absence 8 26 pm

Absence enregistree [19]

absence 2 35 am

Absence enregistree [20]

absence 3 34 am

Absence enregistree [21]

XU-108 XU-110

etudiant 3 21

(3) rebeccah 101 1

- Non-justifiees

[17] 13/am

absence 7 16 pm

Absence enregistree [22]

absence 6 29 pm

Absence enregistree [23]

etudiant 2 39

(2) christer 203 6

- Non-justifiees

[18] 13/am

[2] 17/am

[16] 21/pm

[12] 27/am

[14] 29/pm

[20] 35/am

justificatif 2 18 ma chaine de velo a deraille

Justificatif enregistre

absence 9 10 am

Absence enregistree [24]

justificatif 17 15 j'avais piscine

Justificatif enregistre

absence 5 34 am

Absence enregistree [25]

etudiant 6 40

(6) nicodemus 40 2

- Non-justifiees

[13] 29/am

[23] 29/pm

XU-108 XU-110

absence 9 37 pm

Absence enregistree [26]

absence 2 21 pm

Absence deja connue

absence 4 22 pm

Absence enregistree [27]

etudiant 1 38

(1) hildegard 202 6

- En attente justificatif

[4] 36/pm

- En attente validation

[6] 17/am (ma chaine de velo a deraille)

- Non-justifiees

[10] 3/am

[15] 6/pm

[1] 8/pm

[5] 34/pm

absence 4 8 pm

Absence enregistree [28]

justificatif 19 31 permis de conduire

Justificatif enregistre

inscription evfemia 50

Inscription enregistree (10)

justificatif 20 39 j'avais piscine

Justificatif enregistre

justificatif 19 25 certif. medical

Date incorrecte

absence 4 32 pm

Absence enregistree [29]

absence 2 2 pm

XU-108 XU-110

Absence enregistree [30]

absence 8 33 am

Absence enregistree [31]

justificatif 31 39 certif. medical

Justificatif enregistre

validation 17 ko

Validation enregistree

absence 9 19 pm

Absence enregistree [32]

absence 9 18 am

Absence enregistree [33]

absence 7 27 pm

Absence enregistree [34]

absence 10 20 pm

Absence enregistree [35]

justificatif 7 23 ma chaine de velo a deraille

Date incorrecte

absence 8 18 am

Absence enregistree [36]

absence 7 20 am

Absence enregistree [37]

absence 7 16 am

Absence enregistree [38]

absence 1 4 am

Absence enregistree [39]

absence 10 31 am

Absence enregistree [40]

absence 9 1 pm

Absence enregistree [41]

justificatif 1 13 j'avais piscine

XU-108 XU-110

Justificatif enregistré

absence 6 17 am

Absence enregistrée [42]

justificatif 27 24 j'avais d'autres choses à faire

Justificatif enregistré

absence 2 34 am

Absence enregistrée [43]

justificatif 36 19 j'avais piscine

Justificatif enregistré

justificatif 9 29 convoc. tribunal

Date incorrecte

validation 36 ko

Validation enregistrée

justificatif 14 33 j'aime pas ce cours

Justificatif enregistré

absence 5 37 pm

Absence enregistrée [44]

absence 7 24 pm

Absence enregistrée [45]

absence 10 26 pm

Absence enregistrée [46]

justificatif 33 21 ma chaîne de vélo a dérailé

Justificatif enregistré

étudiants 28

(4) dakota 30 2

(6) nicodemus 40 1

(10) evfemia 50 2

(7) giselle 101 6

(3) rebeccah 101 2

(9) savino 101 4

XU-108 XU-110

(1) hildegard 202 5

(5) anselma 203 1

(2) christer 203 5

(8) salomea 203 2

validation 27 ko

Validation enregistree

justificatif 28 9 ma chaine de velo a deraille

Justificatif enregistre

etudiant 7 22

(7) giselle 101 4

- En attente justificatif

[37] 20/am

- En attente validation

[8] 3/pm (convoc. tribunal)

- Non-justifiees

[38] 16/am

[22] 16/pm

absence 10 4 pm

Absence enregistree [47]

justificatif 23 31 certif. medical

Justificatif enregistre

absence 9 17 am

Absence enregistree [48]

absence 10 7 pm

Absence enregistree [49]

validation 23 ko

Validation enregistree

absence 5 12 pm

Absence enregistree [50]

absence 9 39 am

XU-108 XU-110

Absence enregistree [51]

absence 10 18 am

Absence enregistree [52]

absence 9 36 pm

Absence enregistree [53]

absence 6 33 am

Absence enregistree [54]

absence 2 26 pm

Absence enregistree [55]

absence 1 6 pm

Absence deja connue

validation 33 ko

Validation enregistree

justificatif 21 33 convoc. police

Date incorrecte

validation 2 ko

Validation enregistree

etudiant 9 29

(9) savino 101 5

- Non-justifiees

[41] 1/pm

[24] 10/am

[48] 17/am

[33] 18/am (ma chaine de velo a deraille)

[32] 19/pm

justificatif 48 18 permis de conduire

Justificatif enregistre

absence 9 9 am

Absence enregistree [56]

etudiants 5

XU-108 XU-110

(4) dakota 30 0

(6) nicodemus 40 0

(10) evfemia 50 1

(7) giselle 101 1

(3) rebeccah 101 0

(9) savino 101 1

(1) hildegard 202 2

(5) anselma 203 0

(2) christer 203 1

(8) salomea 203 0

absence 7 13 am

Absence enregistree [57]

justificatif 9 32 convoc. police

Justificatif enregistre

validation 28 ko

Validation enregistree

absence 7 17 pm

Absence enregistree [58]

justificatif 24 13 convoc. police

Justificatif enregistre

justificatif 30 7 certif. medical

Justificatif enregistre

absence 1 5 am

Absence enregistree [59]

justificatif 47 6 Journee defense et citoyennete

Justificatif enregistre

absence 1 7 am

Absence enregistree [60]

justificatif 11 35 j'avais d'autres choses a faire

Justificatif enregistre

XU-108 XU-110

etudiant 5 29

(5) anselma 203 2

- Non-justifiees

[50] 12/pm

[7] 24/am

validation 47 ko

Validation enregistree

justificatif 47 3 ma chaine de velo a deraille

Date incorrecte

absence 2 7 pm

Absence enregistree [61]

validation 24 ok

Validation enregistree

justificatif 15 7 j'avais piscine

Justificatif enregistre

absence 5 9 pm

Absence enregistree [62]

validation 9 ok

Validation enregistree

justificatif 54 37 j'aime pas ce cours

Justificatif enregistre

absence 6 26 pm

Absence enregistree [63]

absence 6 1 am

Absence enregistree [64]

validation 48 ko

Validation enregistree

absence 9 5 pm

Absence enregistree [65]

absence 6 21 am

XU-108 XU-110

Absence enregistrée [66]

justificatif 32 20 convoc. police

Justificatif enregistré

justificatif 65 7 convoc. police

Justificatif enregistré

justificatif 64 4 j'aime pas ce prof

Justificatif enregistré

validation 6 ok

Validation enregistrée

justificatif 53 39 convoc. tribunal

Justificatif enregistré

absence 5 19 pm

Absence enregistrée [67]

absence 5 33 am

Absence enregistrée [68]

justificatif 5 35 j'aime pas ce cours

Justificatif enregistré

absence 1 9 pm

Absence enregistrée [69]

absence 1 19 am

Absence enregistrée [70]

justificatif 59 6 certif. medical

Justificatif enregistré

absence 5 27 pm

Absence enregistrée [71]

validation 32 ko

Validation enregistrée

absence 1 1 pm

Absence enregistrée [72]

validation 65 ko

XU-108 XU-110

Validation enregistree

etudiant 10 12

(10) evfemia 50 2

- Non-justifiees

[47] 4/pm (Journee defense et citoyennete)

[49] 7/pm

absence 5 14 pm

Absence enregistree [73]

justificatif 22 22 certif. medical

Justificatif enregistre

validation 64 ko

Validation enregistree

justificatif 10 5 j'aime pas ce prof

Justificatif enregistre

justificatif 51 45 permis de conduire

Justificatif enregistre

absence 5 29 pm

Absence enregistree [74]

justificatif 44 43 j'aime pas ce cours

Justificatif enregistre

justificatif 62 14 j'avais piscine

Justificatif enregistre

justificatif 42 18 j'aime pas ce cours

Justificatif enregistre

validation 53 ok

Validation enregistree

justificatif 37 26 j'avais d'autres choses a faire

Justificatif enregistre

justificatif 57 17 ma chaine de velo a deraille

Justificatif enregistre

XU-108 XU-110

validation 8 ko

Validation enregistree

absence 6 36 am

Absence enregistree [75]

validation 5 ko

Validation enregistree

absence 9 11 pm

Absence enregistree [76]

justificatif 76 14 convoc. police

Justificatif enregistre

justificatif 38 19 j'aime pas ce prof

Justificatif enregistre

justificatif 13 33 certif. medical

Justificatif enregistre

validation 76 ko

Validation enregistree

validation 42 ko

Validation enregistree

absence 7 32 pm

Absence enregistree [77]

justificatif 66 25 permis de conduire

Justificatif enregistre

absence 6 12 am

Absence enregistree [78]

justificatif 74 35 convoc. tribunal

Justificatif enregistre

justificatif 63 29 j'avais piscine

Justificatif enregistre

absence 7 29 am

Absence enregistree [79]

XU-108 XU-110

absence 5 27 am

Absence enregistrée [80]

justificatif 77 36 convoc. tribunal

Justificatif enregistré

justificatif 71 31 permis de conduire

Justificatif enregistré

absence 7 13 pm

Absence enregistrée [81]

justificatif 78 17 ma chaîne de vélo a dérailé

Justificatif enregistré

justificatif 80 33 convoc. police

Justificatif enregistré

justificatif 4 39 permis de conduire

Justificatif enregistré

validation 63 ok

Validation enregistrée

validation 38 ok

Validation enregistrée

justificatif 39 9 convoc. police

Justificatif enregistré

validation 15 ko

Validation enregistrée

justificatif 79 33 j'aime pas ce cours

Justificatif enregistré

validation 59 ko

Validation enregistrée

justificatif 81 14 Journée défense et citoyenneté

Justificatif enregistré

justificatif 70 23 Journée défense et citoyenneté

Justificatif enregistré

XU-108 XU-110

validation 10 ko

Validation enregistree

validation 81 ko

Validation enregistree

validation 4 ok

Validation enregistree

etudiants 12

(4) dakota 30 1

(6) nicodemus 40 2

(10) evfemia 50 2

(7) giselle 101 1

(3) rebeccah 101 0

(9) savino 101 5

(1) hildegard 202 8

(5) anselma 203 2

(2) christer 203 2

(8) salomea 203 0

etudiants 1

(4) dakota 30 0

(6) nicodemus 40 1

(10) evfemia 50 0

(7) giselle 101 0

(3) rebeccah 101 0

(9) savino 101 1

(1) hildegard 202 1

(5) anselma 203 0

(2) christer 203 0

(8) salomea 203 0

etudiants 41

(4) dakota 30 3

XU-108 XU-110

(6) nicodemus 40 9

(10) evfemia 50 6

(7) giselle 101 12

(3) rebeccah 101 3

(9) savino 101 11

(1) hildegard 202 12

(5) anselma 203 12

(2) christer 203 10

(8) salomea 203 3

validations

Aucune validation en attente

etudiant 1 1

(1) hildegard 202 1

- En attente justificatif

[72] 1/pm

etudiant 1 45

(1) hildegard 202 12

- Justifiees

[6] 17/am (ma chaine de velo a deraille)

[4] 36/pm (permis de conduire)

- Non-justifiees

[72] 1/pm

[10] 3/am (j'aime pas ce prof)

[39] 4/am (convoc. police)

[59] 5/am (certif. medical)

[15] 6/pm (j'avais piscine)

[60] 7/am

[1] 8/pm (j'avais piscine)

[69] 9/pm

[70] 19/am (Journee defense et citoyennete)

XU-108 XU-110

[5] 34/pm (j'aime pas ce cours)

etudiant 2 1

(2) christer 203 0

etudiant 2 45

(2) christer 203 10

- Non-justifiees

[30] 2/pm (certif. medical)

[61] 7/pm

[18] 13/am

[2] 17/am (ma chaine de velo a derailleur)

[16] 21/pm

[55] 26/pm

[12] 27/am

[14] 29/pm (j'aime pas ce cours)

[43] 34/am

[20] 35/am (j'avais piscine)

etudiant 3 1

(3) rebeccah 101 0

etudiant 3 45

(3) rebeccah 101 3

- Non-justifiees

[17] 13/am (j'avais piscine)

[3] 25/am

[21] 34/am

etudiant 4 1

(4) dakota 30 0

etudiant 4 45

(4) dakota 30 3

- Non-justifiees

[28] 8/pm (ma chaine de velo a derailleur)

XU-108 XU-110

[27] 22/pm (j'avais d'autres choses a faire)

[29] 32/pm

etudiant 5 1

(5) anselma 203 0

etudiant 5 45

(5) anselma 203 12

- Non-justifiees

[62] 9/pm (j'avais piscine)

[50] 12/pm

[73] 14/pm

[67] 19/pm

[7] 24/am

[80] 27/am (convoc. police)

[71] 27/pm (permis de conduire)

[74] 29/pm (convoc. tribunal)

[11] 30/pm (j'avais d'autres choses a faire)

[68] 33/am

[25] 34/am

[44] 37/pm (j'aime pas ce cours)

etudiant 6 1

(6) nicodemus 40 1

- En attente justificatif

[64] 1/am

etudiant 6 45

(6) nicodemus 40 9

- Justifiees

[63] 26/pm (j'avais piscine)

- Non-justifiees

[64] 1/am (j'aime pas ce prof)

[78] 12/am (ma chaine de velo a deraille)

XU-108 XU-110

[42] 17/am (j'aime pas ce cours)

[66] 21/am (permis de conduire)

[13] 29/am (certif. medical)

[23] 29/pm (certif. medical)

[54] 33/am (j'aime pas ce cours)

[75] 36/am

etudiant 7 1

(7) giselle 101 0

etudiant 7 45

(7) giselle 101 12

- Justifiees

[38] 16/am (j'aime pas ce prof)

[9] 30/pm (convoc. police)

- Non-justifiees

[8] 3/pm (convoc. tribunal)

[57] 13/am (ma chaine de velo a deraille)

[81] 13/pm (Journee defense et citoyennete)

[22] 16/pm (certif. medical)

[58] 17/pm

[37] 20/am (j'avais d'autres choses a faire)

[45] 24/pm

[34] 27/pm

[79] 29/am (j'aime pas ce cours)

[77] 32/pm (convoc. tribunal)

etudiant 8 1

(8) salomea 203 0

etudiant 8 45

(8) salomea 203 3

- Non-justifiees

[36] 18/am (j'avais piscine)

XU-108 XU-110

[19] 26/pm (permis de conduire)

[31] 33/am (certif. medical)

etudiant 9 1

(9) savino 101 1

- En attente justificatif

[41] 1/pm

etudiant 9 45

(9) savino 101 11

- Justifiees

[24] 10/am (convoc. police)

[53] 36/pm (convoc. tribunal)

- Non-justifiees

[41] 1/pm

[65] 5/pm (convoc. police)

[56] 9/am

[76] 11/pm (convoc. police)

[48] 17/am (permis de conduire)

[33] 18/am (ma chaine de velo a deraille)

[32] 19/pm (convoc. police)

[26] 37/pm

[51] 39/am (permis de conduire)

etudiant 10 1

(10) evfemia 50 0

etudiant 10 45

(10) evfemia 50 6

- Non-justifiees

[47] 4/pm (Journee defense et citoyennete)

[49] 7/pm

[52] 18/am

[35] 20/pm

XU-108 XU-110

[46] 26/pm

[40] 31/am

exit

C:\Users\sarah\OneDrive\Bureau\C\x64\Debug\C.exe (processus 8900) s'est terminé avec le code 0 (0x0).

Appuyez sur une touche pour fermer cette fenêtre. . .

XU-108 XU-110

XU-108 XU-110

XU-108 XU-110

XU-108 XU-110

XU-108 XU-110

XU-108 XU-110