void supprimer_mineurs (liste * L, unsigned age_limite) { /* 4 points */
* Supprime tous les candidats de moins de age_limite années de la liste triée L.*/
void afficher_liste (liste * L) { /* 2 points */
void afficher_liste (liste * L) { /* 2 points * / /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) {
void afficher_liste (liste * L) { /* 2 points */ /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) { /* 2 points */ /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) { /* 2 points * / /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) { /* 2 points */ /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) { /* 2 points */ /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) { /* 2 points */ /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) { /* 2 points */ /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) { /* 2 points */ /* Affiche tous les candidats de la liste L */
void afficher_liste (liste * L) { /* 2 points */ /* Affiche tous les candidats de la liste L */

Institut Supérieur d'Informatique et de Mat	hėmatiques
Examen d'Atelier de Prog. 2	
Classe : L1 Info	A.U. : 2023/2024
Enseignant: Sakka Rouis Taoufik	Durée : 1H 30
Nom et Prénom:	Nombre des Pages : 4
CIN:	Documents Autorisés : Non

Exercice 1:

Dans le cadre de l'organisation d'une compétition sportive, nous souhaitons mettre en place des méthodes permettant aux intéressés de s'inscrire et à l'équipe d'organisation de filtrer ces inscriptions afin d'afficher la liste finale des candidats. Pour ce fait, on vous demande de compléter les fonctions C suivantes.

Indications:

- Traitez tous les cas possibles, y compris les situations où la liste est initialement vide, la suppression du premier élément, du dernier élément ou de tout élément quelconque de la liste.
- La qualité de votre solution, notamment sa complexité, influencera votre note.

```
typedef struct {
        char nom[20];
        unsigned cin;
        unsigned age;
} candidat;

struct cellule {
        candidat cle;
        struct cellule *suivant;
};

typedef struct {
```

struct cellule *premier; struct cellule *dernier;

} liste;

Ne rien écrire ici

struct cellule * rec /* Recherche le correspondante,	candidat c ou NULL si le c	dans la liste andidat n'est	L et ret pas trouvé	tourne é */	un po		
	•••••					 	
	• • • • • • • • • • • • • • • • • • • •					 	
1							
7							
/*Supprime le c recherche précéd	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po ssibles*,	/		
yoid supprimer_ca /*Supprime le c recherche précéc	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po ssibles*,	/		
/*Supprime le c recherche précéd	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po	/ 	 	
/*Supprime le c recherche précéd	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po	/ 	 	
/*Supprime le c recherche précéc	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po	/ 	 	
/*Supprime le c recherche précéd	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po	/ 	 	
/*Supprime le c recherche précéc	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po	/ 		
/*Supprime le c recherche précéd	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po	/ 		
/*Supprime le c recherche précéc	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \ les cas po	Vous po	/ 		
/*Supprime le c recherche précéd	andidat c de dente et veillez	la liste L, s' à traiter tous	il existe. \	Vous po			

void inscription_candidat (liste *L, candidat c) { /* 4 points */ /* Insère un nouveau candidat c dans la liste L. Si un élément de la liste possède le même CIN que le candidat c, l'ancienne candidature doit être supprimée et le nouveau candidat doit être inséré à la bonne position dans la liste.*/
}
void trier_liste (liste * L){