

	Exercice Cours les listes chaînées	
	Matière : programmation procédural 2 Classe(s) : 1A	Unité pédagogique : Algorithmique & Programmation Année universitaire : 2019-2020

On propose de modéliser la gestion des patients dans un cabinet médical.

Un patient est caractérisé par : le nom, le prénom et un champ rdv (pour rendez-vous) de type entier indiquant si le patient a un rendez-vous ou pas : **0** si le patient est sans rendez-vous, **1** si le patient est avec rendez-vous.

Avant d'être consultés par le médecin, les patients sont entrés dans une salle d'attente qui sera modélisée par une **liste chaînée de patients**.

On vous demande de gérer le cabinet médical en le modélisant par une liste chaînée de patients.

- Définir les structures de données **Patient** et **Cellule**.
- Définir le type **liste** comme un pointeur sur la structure Cellule.

Ajouter les prototypes suivants dans **patient.h**, et implémenter ces fonctions.

Partie 1

1. **void saisirPatient(Patient *p);** qui permet de saisir un seul patient.
2. **void afficherPatient(Patient p);** qui permet d'afficher un seul patient.
3. **liste AjoutPatient (liste tete, Patient P)** , qui permet d'ajouter un nouveau patient à la fin de la liste identifiée par son pointeur tete.
4. **void RendezVous (liste tete, int *rdv, int *sansRdv)** , qui compte et retourne le nombre de patients avec rendez-vous, et le nombre de patients sans RDV.

Partie 2

5. **liste SupprimePatient (liste tete)** , qui permet de faire entrer un patient en consultation. Cette opération est effectuée de la manière suivante : s'il n'y a aucun patient avec rendez-vous alors c'est le premier patient de la liste qui est supprimé. Sinon on supprime le premier patient qui a un rendez-vous.
6. **void ConsulterSalleAttente (liste tete)** , qui affiche tout d'abord les patients avec rendez-vous, ensuite les patients sans rendez-vous.
7. **void sauvegarderPatients(liste tete,char *chemin)** , qui permet de sauvegarder les patient dans un fichier accessible selon un chemin
8. **liste recupererListePatients(liste tete,char *chemin)**, qui permet de retourner dans une liste chaînée les données d'un fichier accessible selon un chemin.

Tester ces fonctions dans main.c

NB : La sauvegarde et la lecture des informations à partir d'un fichier sera faite **en mode texte**.