

## Problème : Digitalisation de la gestion des inscriptions sur la plateforme evax.tn (20 points)

Dans le cadre de la stratégie nationale de lutte contre la COVID 19, l'état tunisien se propose de créer une plateforme pour la gestion des réservations pour bénéficier de la vaccination gratuite et prise en charge par l'état. Afin de digitaliser la liste des citoyens, on se propose de créer un annuaire en veillant à l'optimisation de l'espace mémoire utilisé. Chaque citoyen aura une fiche qui contient son nom, le numéro de sa CIN, son numéro de téléphone, sa date de naissance, la date de son inscription sur la plateforme, un champ pour indiquer s'il a déjà eu la COVID, la date de son éventuel test positif ainsi qu'un champ indiquant s'il a des maladies chroniques.

Pour satisfaire ce besoin, nous allons utiliser une liste chaînée en se basant sur les deux structures imbriquées suivantes :

- une structure de données **date** qui contiendra comme champs 3 entiers : **jour**, **mois** et **annee**.
- une structure de données **citoyen** qui représente un inscrit sur la liste et inclut les données :
  - o une chaîne de caractères **nom**, qui correspondra au nom de la personne (**en langage C**, éviter d'utiliser les tableaux statiques pour cette chaîne)
  - o un entier **CIN**, qui correspondra au numéro de la CIN (Carte d'Identité Nationale),
  - o un entier **telephone**, qui correspondra au numéro de téléphone de la personne,
  - o une structure de données **dateNaiss** de type **date**, qui correspondra à la date à laquelle a été rentrée la fiche du citoyen dans la liste chaînée.
  - o une structure de données **dateSaisie** de type **date**, qui correspondra à la date à laquelle a été rentrée la fiche du citoyen dans la liste chaînée.
  - o Une champ booléen ou entier **covid** pour indiquer si le citoyen a déjà eu la COVID
  - o une structure de données **dateCovid** de type **date**, qui correspondra à la date à laquelle le citoyen a été, éventuellement, testé positif à la COVID.
  - o Une champ booléen ou entier **autresMaladies** pour indiquer si le citoyen a des maladies chroniques
  - o un pointeur sur la personne suivante, nommé **suivant**, qui pointera sur l'élément suivant de la liste chaînée.

**NB :** Toutes les questions doivent être écrites en version Algorithmique et traduites en langage C

### Partie 1 : Gestion des inscriptions

- 1- Ecrire les déclarations des structures de données **citoyen** et **date**.
- 2- Ecrire l'instruction qui permet de réserver la mémoire nécessaire pour un élément de type **citoyen**
- 3- Pour assurer une égalité des chances entre les citoyens, le traitement des demandes d'inscriptions respecte la contrainte « FIFO - Premier Arrivé, Premier Servi ». Expliciter la structure de données pouvant le mieux représenter cette liste chaînée.
- 4- Ecrire un sous-programme (module) **AjouteCitoyen** qui permet à l'utilisateur de saisir au clavier les différents champs d'un élément de type **citoyen** et qui l'ajoutera dans la liste.  
**NB :** La liste ne contiendra aucun doublon de **CIN**. Si le numéro CIN figure déjà dans la liste, aucune insertion ne sera faite.
- 5- Ecrire un sous-programme (module) **RechercheCitoyen** pour chercher une personne dans la liste chaînée selon son numéro de **cin** qui renverra l'adresse de la structure correspondante au **cin** ou **NULL** si le nom n'appartient pas à la liste chaînée.
- 6- Ecrire un sous-programme (module) **ProportionCOVID** qui permet à l'utilisateur de déterminer automatiquement la proportion (le pourcentage) des citoyens qui ont déjà eu la COVID parmi la liste des inscrits.

### Partie 2 : Gestion des RDVs

Dans cette partie, nous nous proposons de créer automatiquement la liste des RDVs aux citoyens déjà inscrits en respectant les règles suivantes lors de la création de la nouvelle liste chaînée des RDVs des plus prioritaires aux moins prioritaires (on se limitera aux règles suivantes) :

- Les personnes âgées nées avant 1960
  - Les personnes nées avant 1970 et ayant des maladies chroniques
  - Les personnes nées avant 1970, sans maladies chroniques mais n'ayant pas eu la COVID
  - Les personnes nées avant 1970, sans maladies chroniques et avec un test COVID positif de plus que 06 mois
- 7- Identifier toutes les structures de données nécessaires pour la gestion des RDVs sachant que chaque RDV inclura le numéro de la CIN, la date du RDV, l'heure du RDV et l'adresse.  
**NB :** la liste des RDVs respectera aussi la contrainte « FIFO - Premier Arrivé, Premier Servi »
  - 8- Chaque matin, l'utilisateur lance un sous programme (un module) **majRDV** pour mettre à jour la liste des RDVs pour ajouter les citoyens inscrits la veille à la liste des RDVs et en respectant toutes les contraintes déjà citées.
  - 9- Chaque matin, l'utilisateur lance, aussi, un sous programme (un module) **lyoumRDV** qui permet de lister les RDVs de la journée. Cette liste sera communiquée à l'équipe qui se chargera de vérifier le RDV de chaque citoyen au niveau du centre de vaccination.
  - 10- Après la vaccination d'un citoyen, cette dernière équipe supprimera le RDV de la liste en utilisant le sous-programme **vaccinOK**.

**NB :** Pour faciliter la gestion, nous considérons que tous les citoyens se présenteront dans l'ordre au centre de vaccination