

3- Une fonction « Affichage1 » à deux paramètres TA et NA, permettant d'afficher les informations des adhérents du tableau de structures TA.

4- Une fonction « Affichage2 » à deux paramètres TC et NC, permettant d'afficher les informations des constats du tableau de structures TC.

5- Une fonction « calcul\_total\_remb » à trois paramètres TC, NC et type, permettant de calculer et afficher pour un type de constat donné le montant total des remboursements.



Nom : ..... Prénom : ..... Matricule : .....



DS

## Atelier de Programmation 2

Documents non autorisés

Classe : 1<sup>ère</sup> LI

Enseignant : : TAOUFIK Sakka Rouis & KAIS Ben Salah

Nombre de pages : 6

Durée : 1H00

Calculatrice non autorisée

20

### Exercice 1 : (5 points)

En utilisant la notion de pointeur et le passage par adresse, on vous demande d'écrire une fonction de prototype «void Recherche\_plus\_longue\_seq (int \*T, int n, int \*ind\_debut, int \*taille)» qui permet de chercher la première plus longue séquence d'entiers dans le sens croissant (La séquence contient au minimum deux entiers) dans le tableau T. La séquence d'entiers sera représentée par sa case « ind\_debut » et sa taille « taille ». Si la séquence n'existe pas, la « ind\_debut » et le « taille » seront mis à -1.

Signature  
des  
urveillants



**N.B :** Résoudre cet exercice en utilisant les pointeurs (**sans utiliser les [ ]**) et l'allocation dynamique.  
Une compagnie d'assurances vous a confié le développement d'une application qui permet de gérer les constats de ses adhérents.

**typedef struct**

```
{
    int code;           /* code identifiant de l'adhérent */
    char * nom;         /* Nom de l'adhérent */
    char * prenom;      /* Prénom de l'adhérent */
    float plaf_r;       /* Plafond de remboursement */
}
```

```

} adherent;

```

**typedef struct**

```

int ref;          /* Référence identifiant du constat*/
unsigned type;    /* Type du constat (0: accident de voiture, 1: incendie, 2: vol)*/
float prix;       /* Prix de remboursement */
int code;        /* code identifiant de l'adhérent */

```

```

} constat;

```

On vous demande d'écrire les fonctions C suivantes :

1- Une fonction « Lecture1 » à deux paramètres TA et NA permettant de saisir les informations de **NA** adhérents dans un tableau de structures TA.

2- Une fonction « Lecture2 » à trois paramètres TC, NC et TA, permettant de saisir les informations de NC constats dans un tableau de structures TC. Vérifier l'existence de l'adhérent (**code**) dans le tableau des adhérents **TA**.

**N.B :** Implémenter et utiliser une fonction **TEST** à trois paramètres TA, NA et code qui retourne

- ✓ L'adresse de l'adhérent s'il existe dans TA
- ✓ NULL dans le cas contraire.