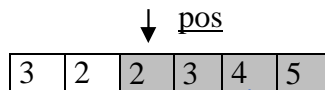


|  |  |  |
|--|--|--|
| Devoir surveillé <input type="checkbox"/>        | Examen <input checked="" type="checkbox"/>   | Session : principale <input checked="" type="checkbox"/><br>de contrôle <input type="checkbox"/> |
| Matière : Algorithmique et structures de données | Semestre: 1  |  |
| Enseignant : Wided Miled et Aymen Sellaouti      | Date: 03/01/2017 à 15h00   |  |
| Filière(s) : MPI                                 | Durée: 1h30  |  |
| Nombre de pages : 2                              | Documents : autorisés <input type="checkbox"/> non autorisés <input checked="" type="checkbox"/> |  |

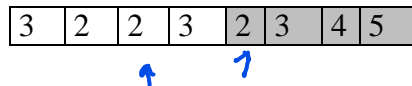
### Exercice 1 (8 pts) :

- 1- Ecrire en **algorithmique** la procédure **decaler()** qui permet de décaler un tableau d'un pas **p** à partir de la position **pos**.

Le tableau T avant le décalage



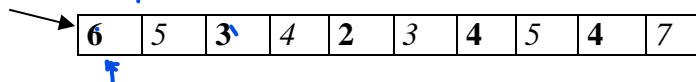
Le tableau T après le décalage d'un pas 2 à partir de la position 3



- 2- Ecrire en **algorithmique** la procédure **remplissage()** qui permet de remplir **p** cases du tableau **T** à partir de la position **pos** avec l'entier **x**.
- 3- Ecrire en **algorithmique** la procédure **decompression()** qui permet de décompresser un tableau d'entiers. La compression consistait à remplacer chaque séquence du même entier par la taille de la séquence suivie de l'entier, comme illustré dans l'exemple ci-dessous, où l'entier est désigné en italique et son nombre de répétition est indiqué en gras. Pour faciliter le travail on suppose qu'un entier se répète au moins 2 fois. La décompression sera le traitement inverse et se fera sur le même tableau et sans utiliser de tableau intermédiaire.

### Exemple:

Le tableau T compressé



Le tableau T décompressé



## Exercice 2 (12 pts) :



Un établissement universitaire dispose des fichiers suivants pour la gestion des notes de ses étudiants :

### « **matiere.txt** »

Chaque ligne de ce fichier contient les données suivantes :

|                  |                         |
|------------------|-------------------------|
| Code matière     | : 5 caractères (entier) |
| Section          | : 10 caractères         |
| Intitulé matière | : 30 caractères         |
| Coefficient      | : 5 caractères (réel)   |

### « **etudiant.txt** »

Chaque ligne de ce fichier contient les données suivantes :

|                |                         |
|----------------|-------------------------|
| Carte étudiant | : 5 caractères (entier) |
| Nom            | : 30 caractères         |
| Prénom         | : 30 caractères         |
| Section        | : 10 caractères         |

### « **notes.txt** »

Chaque ligne de ce fichier contient les données suivantes :

|                |                         |
|----------------|-------------------------|
| Carte étudiant | : 5 caractères (entier) |
| Code matière   | : 5 caractères (entier) |
| Note           | : 5 caractères (réel)   |

La moyenne pour un étudiant d'une section donnée est calculée de la façon suivante :

$$ME = \text{TOTAL\_NOTES} / \text{TOTAL\_COEFF}$$

TOTAL\_NOTES = la somme de chaque note obtenu pour une matière \* le coefficient de cette note.

TOTAL\_COEFF : le total des coefficients de la section.

On vous demande d'écrire les fonctions suivantes :

1- float **get\_coeff\_matiere**(FILE \*fp, int code\_matiere){ }  
qui retourne le coefficient d'une matière.

2- float **total\_coeff\_section** (FILE \*fp, char \*section){ }  
qui retourne le total des coefficients des matières d'une section.

3- float **moyenne\_etudiant** (FILE \*fp\_matiere, FILE \*fp\_note, int carte\_etudiant, char \*section\_etudiant) { }  
qui retourne la moyenne d'un étudiant d'une section donnée.

4- void **actualise\_note\_etudiant**(FILE \*fp, int code\_matiere, int\_carte\_etudiant, float note)  
{ }  
qui permet de corriger la note d'un étudiant dans une matière donnée.

Bonne chance 😊