

Nom :
Prénom :
Groupe :

C1-N1-AC2 : Concevoir une solution logicielle (Algorithmie/C++)

Réponses à rédiger sur la feuille double fournie en indiquant vos nom, prénom, groupe IUT ainsi que sur le sujet lorsque c'est explicitement mentionné.

On s'intéresse dans cet exercice à la gestion d'une promotion d'étudiants

Un `Etudiant` sera caractérisé par :

- son nom représenté par une chaîne de caractères (vue comme un tableau statique de caractères) permettant d'afficher jusqu'à 20 caractères imprimables,
- son prénom représenté par une chaîne de caractères (vue comme un tableau statique de caractères) permettant d'afficher jusqu'à 20 caractères imprimables,
- sa moyenne générale de type réel.
- son état de scolarité de type entier (0 pour un étudiant démissionnaire au cours du semestre et 1 pour un étudiant suivant le cursus).

La promotion d'`Etudiant` sera représentée par un tableau `promo` de taille fixée à 70.

Au cours du semestre, certains étudiants ont démissionné. A l'approche des jurys, on souhaiterait ne pas tenir compte de ces étudiants démissionnaires.

On souhaite donc proposer une fonction `Filtre_demission` dont le rôle sera de modifier ce tableau en déplaçant les étudiants démissionnaires en fin de tableau, sans modifier l'ordre des autres étudiants et qui retournera le nombre d'étudiants non démissionnaires.

On donne la structure du lexique et de l'algorithme principal (incomplets) permettant d'appeler les différentes fonctions permettant la gestion de cette promotion.

Lexique principal:

`Etudiant`: 1 type composé de

<

nom:

prenom:

moy:

dem:

>

{

R: Permet à un utilisateur de saisir un certain nombre d'étudiants et de les mémoriser, dans l'ordre alphabétique de leur nom, dans un tableau d'effectif maximal donné par sa taille réelle passée en paramètre

E: 1 entier: la taille réelle du tableau d'`Etudiant`

E/S : 1 tableau d'`Etudiant`

S: 1 entier: le nombre d'étudiants saisis par l'utilisateur.

}

Saisie_Promo:

```

{
R: 
E: 
E/S : 
S: 
}
Filtre_demission: 

```

```

{
R: Calcule et retourne la moyenne générale d'une promotion d'étudiants
E: 1 entier: le nombre d'étudiants présentés au jury
    1 tableau d'Etudiant: correspondant aux étudiants de la promotion
E/S: vide
S: 1 réel: la moyenne des étudiants non démissionnaires.
}

```

Moyenne:

TAILLE_MAX:

promo:

nb_etu_initial, nb_etu_jury:

moyenne_promo: 1 réel

Algo principal:

```

Début
    nb_etu_initial ← Saisie_promo (  )
    nb_etu_jury ← Filtre_demission (  )
    moyenne_promo ← Moyenne (  )
    Ecrire("La moyenne de la promotion est de: ", moyenne_promo)

```

Fin

Questions :

1. Compléter, en algorithmie et **directement sur le sujet** dans les cadres appropriés, les parties manquantes du lexique principal et de l'algorithme principal.
2. Sur votre feuille de composition, donner la déclaration en C++ du type **Etudiant** en précisant dans quel fichier se fait cette déclaration.
3. Sur votre feuille de composition, donner directement en C++, la déclaration de la fonction **Moyenne**.
4. Sur votre feuille de composition, donner directement en C++, la définition de la fonction **Filtre_demission**.

On ne demande pas la déclaration en C++ ni la définition (Algo ou C++) de Saisie_promo.