

TD 10 - Tableaux (2)

1. Tableaux de caractères

On veut construire l'adresse électronique d'un étudiant de l'UTC à partir de son prénom et de son nom. Par exemple l'étudiant Jean Dupont aura l'adresse Jean.Dupont@etu.utc.fr

Le nom et le prénom sont stockés dans 2 chaînes de 20 caractères, l'adresse électronique sera stockée dans une chaîne de 40 caractères.

Les règles de construction d'adresse sont les suivantes :

- Une adresse est composée du prénom suivi d'un point (.) suivi du nom et enfin de l'extension @etu.utc.fr
- On n'accorde pas d'importance aux majuscules et minuscules.
- Dans le cas d'un prénom et d'un nom composés d'un grand nombre de caractères, c'est le prénom qui est tronqué. Par exemple l'étudiante Marie-Sebastienne Dumoulinabord aura l'adresse Marie-Sebastie.Dumoulinabord@etu.utc.fr
- Le caractère d'espacement n'étant pas autorisé dans les adresses électroniques, il sera automatiquement omis. Par exemple l'étudiant Pierre De La Tourdenface aura l'adresse Pierre.DeLaTourdenface@etu.utc.fr

1.1 Ecrire la fonction *construitAdresseMail(prenom, nom, adresse)* qui construit l'adresse électronique d'un étudiant à partir de *prenom* et *nom*.

1.2 Ecrire le programme qui demande à l'utilisateur son nom et son prénom, construit l'adresse électronique et l'affiche

2. La bataille navale

Ecrire un programme qui simule une bataille navale. La grille sera stockée dans un tableau à deux dimensions (5 sur 5). L'ordinateur placera 5 bateaux à l'intérieur de la grille, pour cela on utilisera la fonction *rand()* pour calculer les indices *i* et *j* correspondant à la ligne et la colonne du bateau à placer. Si un bateau se trouve déjà sur l'emplacement l'ordinateur redonnera une autre position. L'utilisateur devra « deviner » l'emplacement des bateaux. Le programme s'arrêtera lorsque tous les bateaux auront été trouvés, on affichera alors le nombre de coups qu'il aura été nécessaire pour trouver les bateaux.

3. Fusion de deux tableaux

Ecrire une fonction *fusionTab()* qui prend en entrée deux tableaux *tab1* et *tab2* (de dimensions respectives *N* et *M*), triés par ordre croissant, fusionne les éléments de *tab1* et *tab2* dans un troisième tableau *tabFus* trié par ordre croissant. On exploitera le fait que *tab1* et *tab2* sont déjà triés pour construire *tabFus*.